NASA CR
147757

28234–6029–RU–00

(NASA-CR-147757) LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION, VOLUME 3 (TRW SYSTEMS GROUP) 611 P HC $16.25 CSCL 02C

N76-25632

UNCLAS
G3/43 42118

LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION

VOLUME III

MAY 1976

LARGE AREA CROP INVENTORY EXPERIMENT

LACIE

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lyndon B. Johnson Space Center
Houston, Texas

Contract Number NAS-9-14547

TRW
SYSTEMS GROUP
ONE SPACE PARK · REDONDO BEACH · CALIFORNIA 90278

28234–6029–RU–00

LACIE PERFORMANCE PREDICTOR FINAL OPERATIONAL CAPABILITY PROGRAM DESCRIPTION

VOLUME III

MAY 1976

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lyndon B. Johnson Space Center
Houston, Texas

Contract Number NAS-9-14547

TRW
SYSTEMS GROUP
ONE SPACE PARK · REDONDO BEACH · CALIFORNIA 90278

This document, in three volumes, describes the FOC version of the LACIE Performance Predictor produced under Contract NAS9-14547. NASA/JSC Contract Technical Monitor is I. D. Browne, Earth Observations Division.

Table of Contents

LEM BOOK VIII

Table of Contents

PART I

PROBLEM DESCRIPTION FOR THE

LEM PROGRAM

INCLUDING

LEM, CAMS, CAS AND YES SUBPROGRAMS

PART I

PROBLEM DESCRIPTION FOR THE LEM SUBPROGRAM

Table of Contents

Problem Description for the LEM Program

1.0 SCOPE

This document describes the requirements and processing logic for the LACIE Error Model program (LEM). This program is an integral part of the Large Area Crop Inventory Experiment (LACIE) system.

1.1 PROGRAM CAPABILITIES

LEM is that portion of the LPP (LACIE Performance Predictor) which simulates the sample segment classification, strata yield estimation, and production aggregation. LEM controls repetitive Monte Carlo trials based on input error distributions to obtain statistical estimates of the wheat area, yield, and production at different levels of aggregation. LEM interfaces with the rest of the LPP through a set of data files. The input files for LEM consist of the following:

- Segment ID file
- Crop Calendar file
- Substrata Historical file
- CAMS Error Model file
- YES Error Model file
- Signature Extension file
- Data Acquisition file

The program generates two output files for use by the output processor.

In addition, three intermediate files (Segment Truth file, CAMS Output file, and YES Output file) are generated by various modules within LEM and may be saved for subsequent runs on the LEM program permitting the bypassing of specified modules on those runs.

1.2 PROGRAM DEVELOPMENT AND ORGANIZATION

The program will be initially developed on the CDC 6600 Timesharing system and later converted to the UNIVAC 1108 under Exec II and Exec VIII. In order to make program development, modification, check out, and conversion easier, the following guidelines will be observed:

a. Modular programming techniques will be used. In particular, the LEM program will contain the following subprograms:

   - Segment Truth Generator
   - CAMS Simulator
   - YES Simulator
   - CAS Simulation

b. All error processing will be through a standard error routine ERRMES.

c. The program will be coded in ANSI Standard FORTRAN. To provide for compatibility between the CDC 6600 and the UNIVAC 1108, the "Rules for Program Development" written by G. Hull for the LACIE Project will be faithfully observed.

d. In order to clarify the coding, comments will be used extensively throughout the program. In particular:

   - Each local quantity will be described within the subroutine using it.
   - Comments will be used to relate the coding to the Problem Definition and Flow Charts.

1.3 OPERATIONAL ASSUMPTIONS

- Only 1 case may be run at a time.
- Only 1 country may be considered in a case.
- A maximum of 999 Monte Carlo trials may be run in a case and a maximum of 100 trials may be executed on any given run.
- A maximum of 4 crop calendar windows and 14 additional prediction points can be processed.
- A maximum of 10 regions per country can be processed.
- A maximum of 50 zones per country can be processed.
- A maximum of 20 strata per zone can be processed.
- A maximum of 325 strata per country can be processed.
- A maximum of 60 substrata per strata can be processed.
- A maximum of 3200 substrata per country can be processed.
- A maximum of 300 substrata per zone can be processed.
- A maximum of 4000 segments per country can be processed.
- A maximum of 300 acquired segments per zone can be processed.
- A maximum of 4000 segments per country can be processed.
- All control card input data will be echo printed.
- All control card input data will be checked for errors before any error will cause the processing of a case to terminate.
- In a repetitive Monte Carlo trial case, normally the individual subprogram reports will be allowed to print during the first and last trial only. An option will exist to eliminate all reports or allow printing of all reports for each trial or for just the last trial.
- All input data files will be checked for correct case numbers.
- The program will require less than 20,000 words of storage in the CPU of the UNIVAC 1108.
- The program will have a restart capability which will allow the program to continue with additional Monte Carlo iterations starting from the last iteration of the previous run.
- All files will be in country, region, zone, stratum, substratum and segment order (to whatever level that is appropriate).
- A maximum of 10 classes will exist in a zone.

2.0 INPUT

There are one control card set and up to seven files required for input to the LEM program. The control card set specifies the problem title, the initial random number seeds, and various program control flags. The following files may be generated by other programs within the LACIE system and input to LEM:

- Segment ID file
- Crop Calendar file
- Substrata Historical file
- CAMS Error Model file
- YES Error Model file
- Signature Extension file
- Data Acquisition file

In addition, the following files may be generated by LEM on one run and then saved and input back into LEM on a subsequent run:

- Segment Truth file
- CAMS Output file
- YES Output file
- CAS Cumulative Output File
- CAS Distribution Output File

2.1 CARD INPUT

2.1.1 List of Data Quantities

See Input Data Description sheet on the following pages.

2.1.2 Card Formats

The LEM program requires four control cards. Each card has a fixed field format as shown in Figure 2-1.

"LEM" is punched in columns 75-77 of each control card and a sequence number is punched in columns 79-80.

Card 1 Header card: Alphanumeric problem header entered in Columns 1-60. LEM is entered in Columns 75-77 and 01 in Columns 79-80.

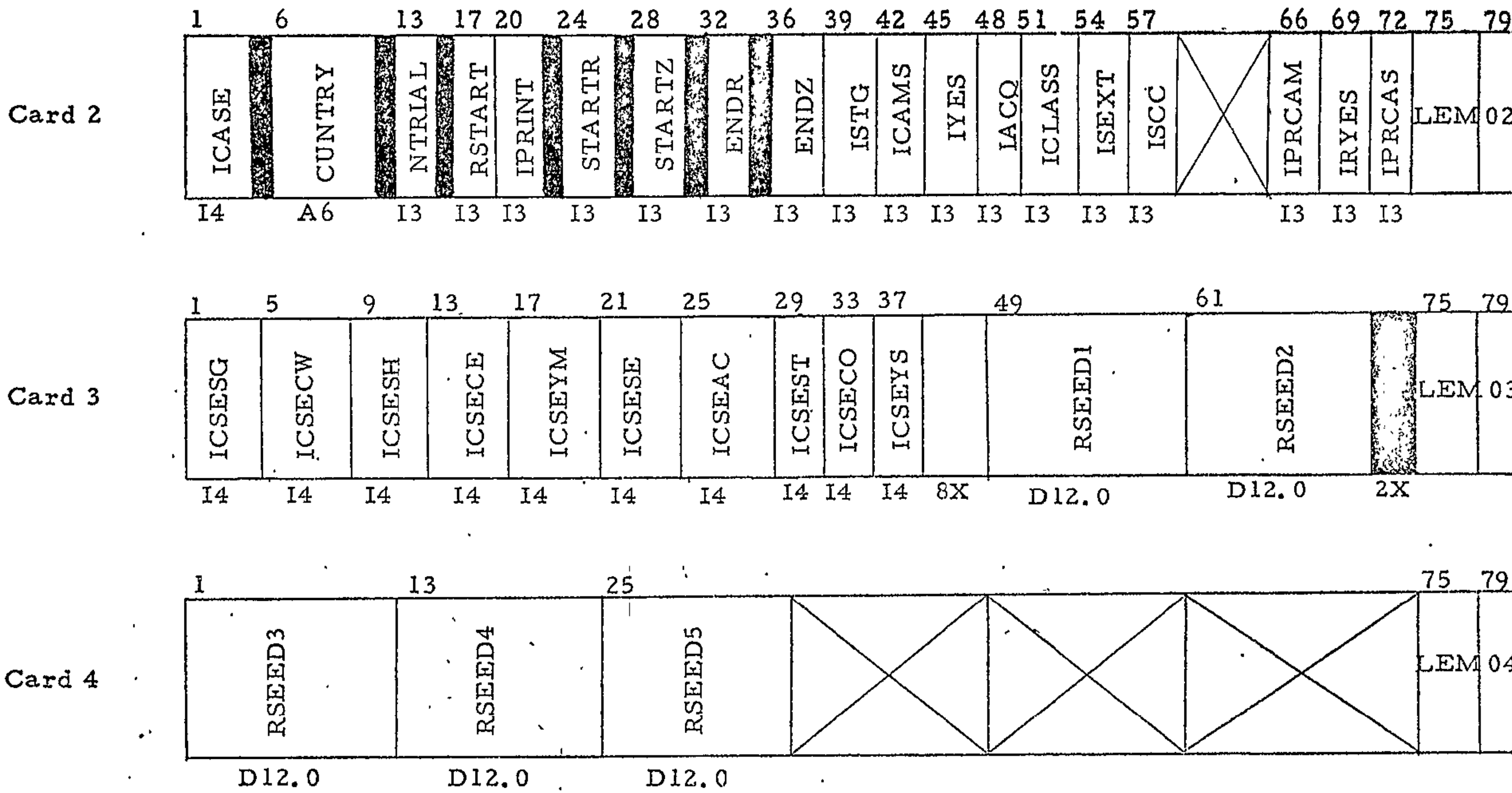


Figure 2-1. Data Card Formats

Card Col.	Name	Dimension	Nominal Value	Range	Description
1-60	TITLE	10	blanks	---	Problem header to be printed at the top of each output page. (format 10A6)
1-4	ICASE	1	0	0-9999	Case number
6-11	CUNTRY	1	--	---	Country (must agree with all input files).
13-15	NTRIAL	1	1	1-999 (NTRIAL-RSTART ≤ 100)	Total number of Monte Carlo iterations at the end of the current run (including previous runs if this is a restart). NTRIAL must be ≤100 if CAS distribution file is to be generated (i.e., if DISTFF ≠ 0 in CAS input)
17-19	RSTART	1	0	0-999	Restart Flag: = n ≠ 0 to restart after n Monte Carlo iterations, 0 if this is not a restart.
20-22	IPRINT	1	0	0-3	Iteration Print Flag (for Segment Truth Generator) 0 to print first and last iterations; 1 to print each iteration; 2 to print last iteration only; 3 to suppress printing.
24-26	STARTR	1	0	0-999	Starting region number. } Both zero or both non-zero
28-30	STARTZ	1	0	0-999	Starting zone number. } Both zero or both non-zero
32-34	ENDR	1	0	0-999	Ending region number. } Both zero or both non-zero
36-38	ENDZ	1	0	0-999	Ending zone number. } Both zero or both non-zero
39-41	ISTG	1	0	0-3	Segment Truth Flag: 0 to vary error statistically, 1 to hold error constant using results from the first iteration only, 2 to hold error constant using a previously generated Segment Truth file, 3 to eliminate the Segment Truth error (error is zero).
42-44	ICAMS	1	0	0-3	CAMS Error Flag: Usage is similar to the usage of the Segment Truth flag described above except that for the case in which ICAMS = 2, the CAMS Output file is used.

Card Col.	Name	Dimension	Nominal Value	Range	Description
45-47	IYES	1	0	0-3	YES Error Flag: Usage is similar to the usage of the Segment Truth flag described above except that for the case in which IYES = 2, the YES Output file is used.
48-50	IACQ	1	0	0-1	Segment Acquisition Flag: 0 to include segment acquisition conditions, 1 to eliminate segment acquisition conditions.
51-53	ICLASS	1	0	0-2	Classification Error Flag: 0 to vary classification error in CAMS, 1 to hold the classification error constant, 2 to eliminate the classification error (set it to zero).
54-56	ISEXT	1	0	0-2	Signature Extension Error Flag: 0 to vary Signature Extension Error, 1 to hold Signature Extension Error constant, 2 to eliminate the Signature Extension Error (error is zero).
57-59	ISCC	1	0	0-2	Segment Crop Calendar Error: 0 to vary the error, 1 to hold the error constant, 2 to eliminate the error (error is zero).
66-68	IPRCAM	1	0	0-3	Iteration print flag for CAMS: Usage is similar to that of IPRINT.
69-71	IPRYES	1	0	0-3	Iteration print flag for YES: Usage is similar to that of IPRINT.
72-74	IPRCAS	1	0	0-3	Iteration print flag for CAS: Usage is similar to that of IPRINT.

Card Col.	Name	Dimension	Nominal Value	Range	Description
1-4	ICSESG	1	0	0-9999	Case number for Segment ID file.
5-8	ICSECW	1	0	0-9999	Case number for Crop Calendar file.
9-12	ICSESH	1	0	0-9999	Case number for Substrata Historical file.
13-16	ICSECE	1	0	0-9999	Case number for CAMS Error Model file.
17-20	ICSEYM	1	0	0-9999	Case number for YES Error Model file.
21-24	ICSESE	1	0	0-9999	Case number for Signature Extension file.
25-28	ICSEAC	1	0	0-9999	Case number for Data Acquisition file.
29-32	ICSEST	1	0	0-9999	Case number for Segment Truth file.
33-36	ICSECO	1	0	0-9999	Case number for CAMS Output file.
37-40	ICSEYS	1	0	0-9999	Case number for YES Output file.
49-60	RSEED1	1	1	1-999999999999	Initial random number seed for Segment Truth Error (odd positive integer in F-format).
61-72	RSEED2	1	1	1-999999999999	Initial random number seed for Classification Error.
1-12	RSEED3	1	1	1-999999999999	Initial random number seed for Signature Extension Error.
13-24	RSEED4	1	1	1-999999999999	Initial random number seed for Segment Crop Calendar Error.
25-36	RSEED5	1	1	1-999999999999	Initial random number seed for Yield Error.

2.1.3 Deck Set Up

Each of the four LEM control cards is required and they must be in card number order. In addition, control cards are always required for the CAMS module even if this module is not used. If CAMS is skipped (e.g., ICAMS = 2), then the corresponding control cards must still be included. Finally, the control cards for CAS must always be specified. The contents and format of the CAMS and CAS control cards are specified in the Problem Descriptions for CAMS and CAS.

2.1.4 Rules for Entering Data on Cards

1. Integers must be right justified.
2. Alphanumeric fields should be left justified.
3. Real (i.e., floating point) fields must have the decimal point present.

2.2 INPUT FILES

The following files may be input to the LEM program:

	Source	Destination
Segment ID File	LUMP	STG
Crop Window (Calendar) File	LUMP	CAMS
Substrata Historical File	LUMP	STG, CAS
CAMS Error Model File	SEE	CAMS
YES Error Model File	SEE	YES
Signature Extension File	SEE	CAMS
Data Acquisition File	SACS	CAMS
Segment Truth File	STG	CAMS
CAMS Output File	CAMS	CAS
YES Output File	YES	CAS
CAS Cumulative File	CAS	CAS
CAS Distribution File	CAS	CAS

Note that the last five files listed above are generated by modules within LEM. They may be saved and input to LEM on subsequent runs.

2.3 RESTRICTIONS

1. In general, the various error sources may be independently varied, held constant, or eliminated by specifying appropriate values for the input parameters ISTG, ICAMS, IYES, ICLASS, ISEXT, ISCC, ICAS2, and ICAS3. However, the user may not specify ISTG = 0 unless ICAMS is also zero. The reason for this restriction is that if ICAMS = 1 or 3 for example, the CAMS Output file will be generated on the first iteration and then used on all subsequent iterations. If ISTG were zero indicating that the Segment Truth error was to be varied, the program would be in trouble because to vary the Segment Truth error means that the results on the CAMS Output file must also be variable even if the CAMS errors are constant or zero. The way around this restriction is to specify both ICAMS = 0 and ISTG = 0 and then to specify non-zero values for ICLASS, ISEXT, and ISCC. In this manner the CAMS Output file will be written on each iteration even though the CAMS errors are really constant.

2. The variable possible combinations of the input parameters ICAMS, ISTG, IYES, ICLASS, ISEXT, and ISCC as well as other options (CAMS classification model, multi-temporal sampling, and acquisition effects) are presented in the chart on the following page.

3. On a restart run the input case number ICASE must agree with the case number on both of the following files which may be input to LEM:

   - CAS Cumulative Output file
   - CAS Distribution Output file

ICAMS	ISEXT	ISCC	ICLASS	Model	Multi-Temp Sampling	ISTG	IYES	IACQ
0	0, 1, 2	0, 1, 2	0, 1, 2	1, 2	0, 1	0, 1, 2, 3	0, 1, 2, 3	0, 1
1	1, 2	1, 2	1, 2	1, 2	0, 1	1, 2, 3		0, 1
2	X	X	X	X	X	X		X
3	X	X	X	1, 2	X	3		0, 1

For ICAMS, ISTG, IYES

0 $\Longrightarrow$ vary error,

1 $\Longrightarrow$ hold error constant; use first iteration results,

2 $\Longrightarrow$ hold error constant; use previously generated file,

3 $\Longrightarrow$ eliminate error.

For ISEXT, ISCC, ICLASS

0 $\Longrightarrow$ vary error,

1 $\Longrightarrow$ hold error constant,

2 $\Longrightarrow$ eliminate error.

For Multi-Temporal Sampling

0 $\Longrightarrow$ include multi-temporal sampling effects,

1 $\Longrightarrow$ exclude effects.

For Acquisition Effects

0 $\Longrightarrow$ include segment acquisition effects,

1 $\Longrightarrow$ eliminate segment acquisition effects.

Model = 1 or 2 for classification model 1 or 2 respectively in CAMS

4. On a restart run the CAS Cumulative Output file and the CAS Distribution Output file must always be specified. In addition, the Segment Truth file, the CAMS Output file and/or the YES Output file should be specified if the input flags ISTG, ICAMS, and/or IYES are set to 1, 2, or 3. Note, however, that if the CAMS Output file is specified, then it is not necessary to specify the Segment Truth file.

3.0 PROCESSING

3.1 OVERVIEW

The LEM program consists of control logic to execute one or more application subprograms as selected by the user through the control card parameters. The effect of various input error distributions is taken into account by use of a Monte Carlo technique employing a random number generator. The application subprograms are as follows:

<u>Segment Truth Generator</u>

This subprogram generates true proportion of wheat and the true proportion of mixed pixels for each sample segment.

<u>CAMS Simulator</u>

This subprogram models the LACIE segment classification process and associated errors to provide an estimate of the proportion of wheat in each satellite acquired sample segment and a measure of the classification error.

<u>YES Simulator</u>

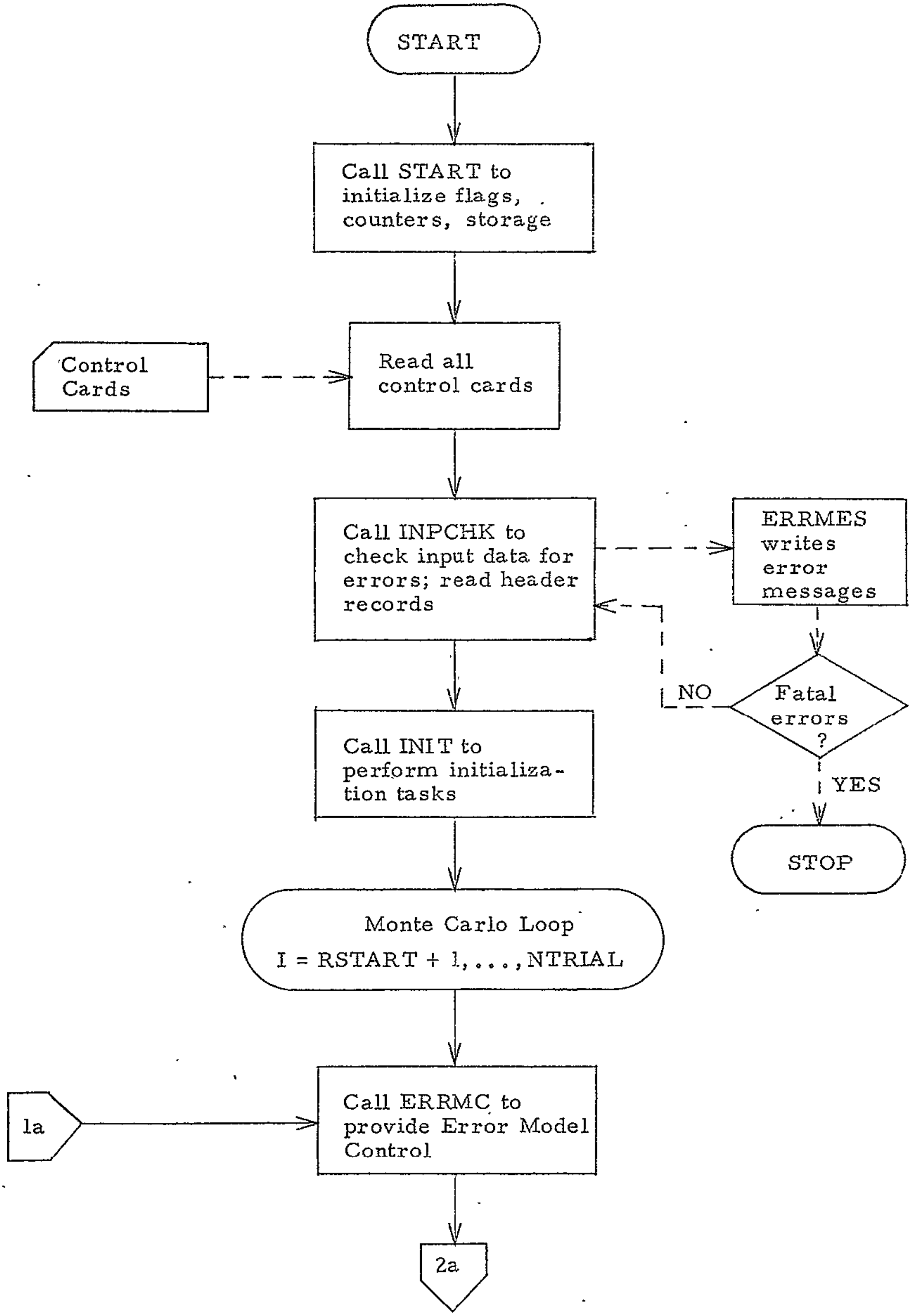
This subprogram models the yield estimation process and associated errors in order to provide an estimate of the wheat yield for each strata and a measure of the yield error.

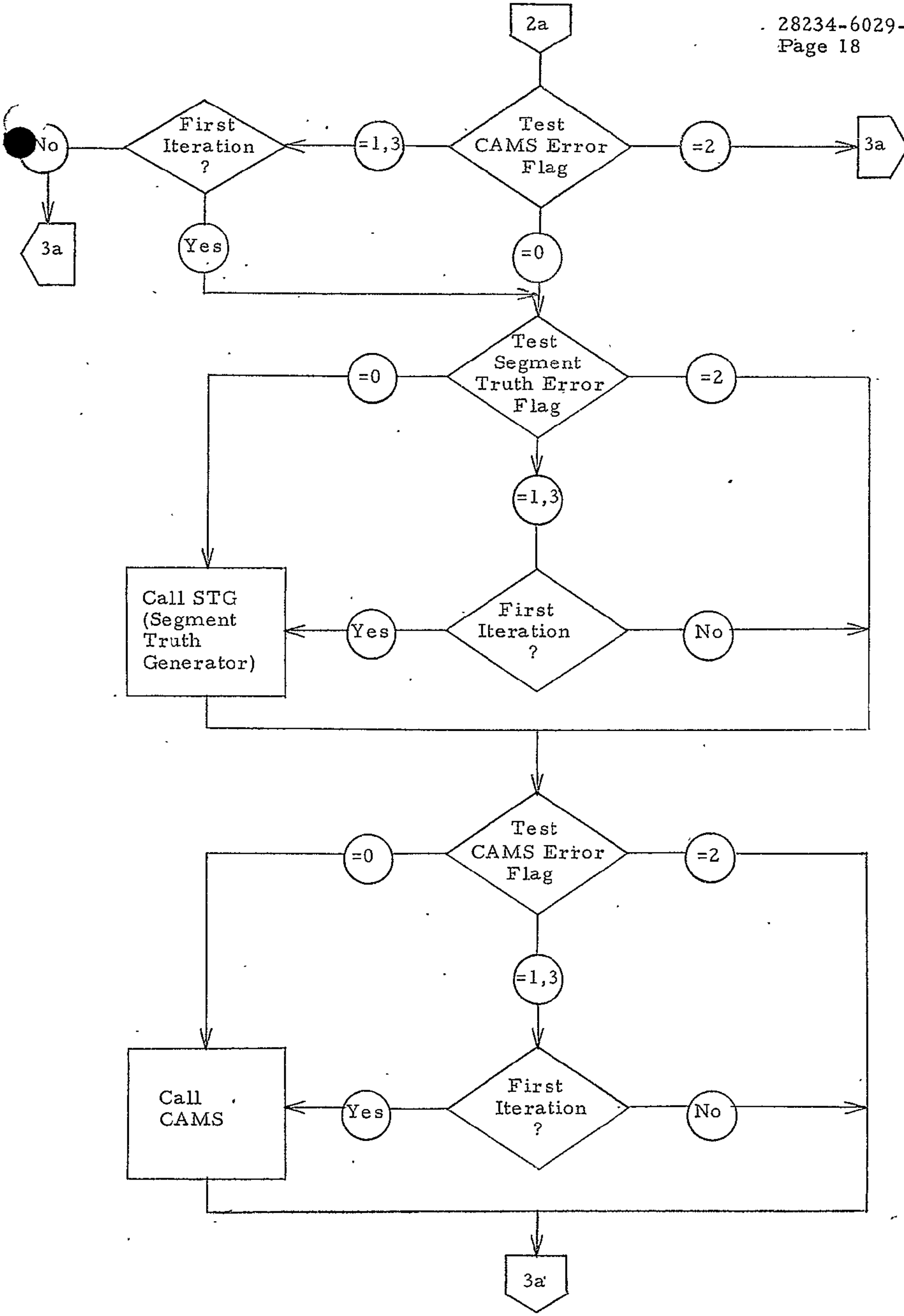
<u>CAS Simulator</u>

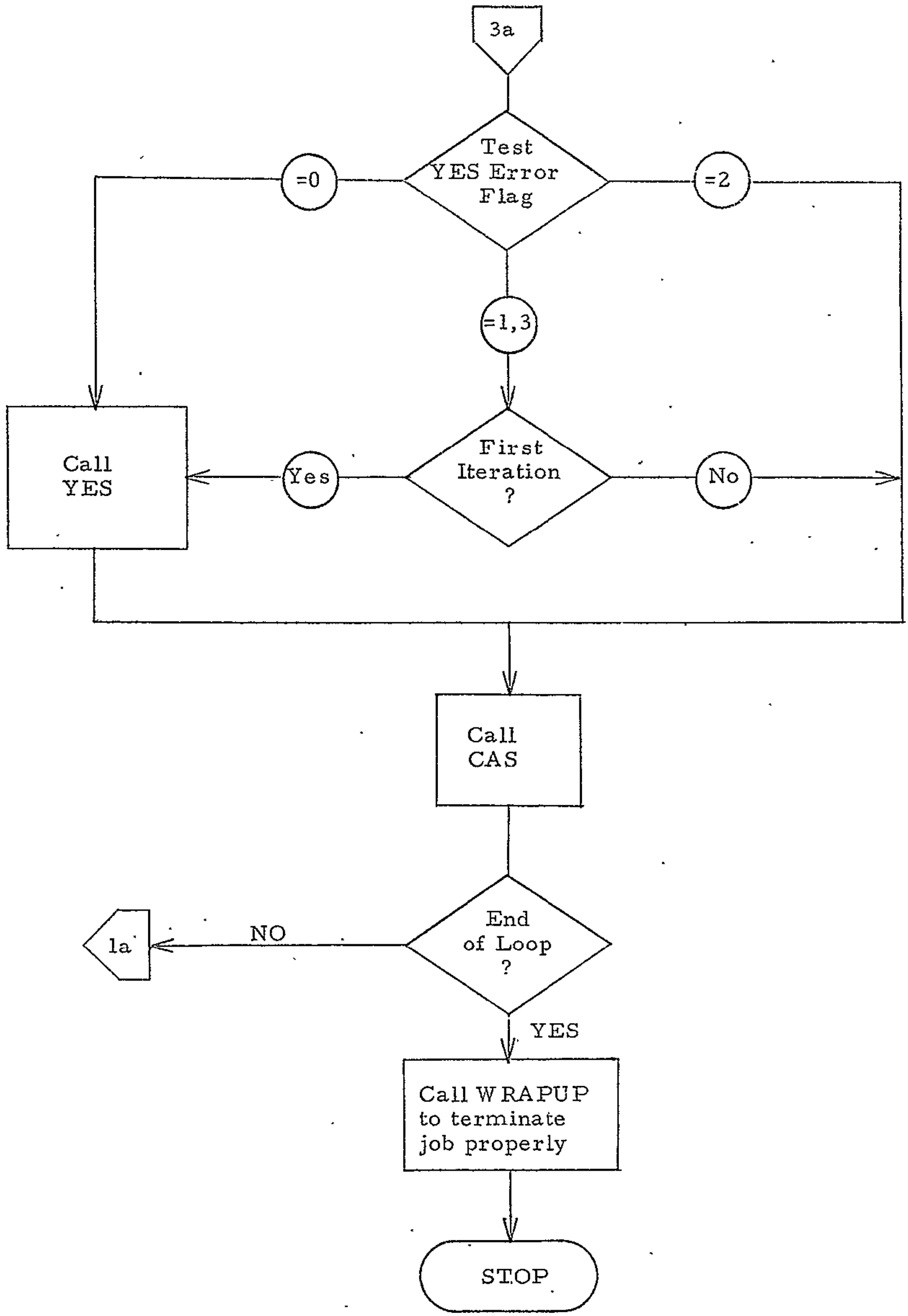
This subprogram models the LACIE aggregation technique including the aggregation of wheat area and production to the country level and the estimation of the accuracy of the aggregation. CAS also compares the estimates to the truth baseline to compute actual errors.

3.2 PROGRAM FLOW

An overall flow diagram of the LEM program is presented on Pages 14-16.


START
Call START to initialize flags, counters, storage
Control Cards
Read all control cards
Call INPCHK to check input data for errors; read header records
ERRMES writes error messages
Fatal errors ?
NO
YES
STOP
Call INIT to perform initialization tasks
Monte Carlo Loop
I = RSTART + 1, . . . , NTRIAL
1a
Call ERRMC to provide Error Model Control
2a


2a
Test
CAMS Error
Flag
=1,3
=2
3a
First
Iteration
?
No
3a
Yes
=0
Test
Segment
Truth Error
Flag
=0
=2
=1,3
Call STG
(Segment
Truth
Generator)
Yes
First
Iteration
?
No
Test
CAMS Error
Flag
=0
=2
=1,3
Call
CAMS
Yes
First
Iteration
?
No
3a


3a
Test YES Error Flag
=0
=2
=1,3
First Iteration ?
Yes
No
Call YES
Call CAS
End of Loop ?
NO
1a
YES
Call WRAPUP to terminate job properly
STOP

3.3 PROCEDURES AND EQUATIONS

3.3.1 Job Initialization

Initialize storage, flags, and counters.

3.3.2 Read All Control Cards

(LEM, CAMS, and CAS)

3.3.3 Input Check

- Check all parameters from control cards for errors.
- Read header records of all input files and compare country and case number from each file to input values.

3.3.4 Monte Carlo Loop

Perform steps for each Monte Carlo trial.

1. CALL ERRMC to properly initialize the random number seeds for the various error sources as specified by input flags.
2. Obtain the Segment Truth data by calling STG or by having CAMS read from the Segment Truth file as specified by the Segment Truth error flag.

   The procedures to be used in the Segment Truth Generator are as follows:

   - Calculate PM_{K_i} by calling BETAD with

     mean = PW_K and

     standard deviation = $PW_K * CV_2$

   - Calculate PM_{K_i} by calling BETAD with

     mean = $PW_{K_i} * (\delta PM)$ and

     standard deviation = $PW_{K_i} * (\delta PM) * CV_3$

δPM = ratio of mixed pixels to true proportion wheat

3. Obtain the CAMS error data by calling CAMS or by having CAS read from the CAMS Output file as specified by the CAMS error flag.

4. Obtain the YES error data by calling YES or by having CAS read from the YES Output file as specified by the YES error flag.

5. Call CAS

3.3.5 Program Termination

After the final Monte Carlo iteration, call WRAPUP to terminate the job properly.

4.0 OUTPUT

The LEM program produces printed reports in the Segment Truth Generator, CAMS, YES, and CAS modules and generates two output files for further processing by the LEM Post Processor. In addition, three intermediate files are generated by the Segment Truth Generator, CAMS, and YES modules. Finally, program status information about each subprogram and LEM itself after each Monte Carlo trial and at the end of the program execution are printed out.

4.1 PRINTED DATA

4.1.1 Printed Reports

Printed reports are generated by the Segment Truth Generator, CAMS, YES, and CAS modules. The Segment Truth Report consists of the substrata true PW, each segment true PW, and each segment true PM plus the average segment true PW for each substratum. The CAMS module generates two reports -- a CAMS estimated proportion wheat summary and an error source report. YES generates a report specifying the true yield of each stratum, and for each prediction point of each stratum, the estimated yield, estimated date, standard error, and percent error. CAS generates a summary report for the LEM simulation. The content and format of the CAMS, YES, and CAS modules will be described in the Problem Definitions for each of those subprograms. The format of the Segment Truth report is as follows:

Segment Truth Report

True Substrata PW	True Segment PW	Ave. PW	Error PW	True Segment PM
XXX	XXX	XXX	XXX	XXX
	XXX			XXX
	XXX			XXX
	⋮			⋮
XXX	XXX	XXX	XXX	XXX
	XXX			XXX
	⋮			⋮

4.1.2 Intermediate Debug

At the present there is no Intermediate Debugging printout specified. However, it is anticipated that during checkout the contents of various files will be printed out as the data records are written.

4.1.3 Status Information

At the conclusion of the run the following status information is printed out:

- Number of data records read from each input file
- Number of non-fatal errors detected in the input data
- Number of non-fatal errors detected during execution
- Number of Monte Carlo trials completed
- Final random number seeds (to be input on restart run)
- Number of data records written on each output file

4.1.4 Echo Print Input Card Images

The data specified on the input control cards is always printed out in a format that is similar to the format on the input card images. Due to differences in the FORTRAN read and write formats, the printout may be slightly different from the input card images. For example, a blank field will be printed out as -0 rather than being left blank.

4.2 FILES

There are two output files generated by LEM -- the CAS Cumulative Output File and the CAS Distribution Data File. In addition, intermediate files are generated by the Segment Truth Generator, CAMS, and YES. These intermediate files may be saved and input on subsequent runs allowing specified subprograms to be bypassed provided the corresponding error contributions are constant.

4.2.1 CAS Cumulative Output File

This file contains the cumulative information being generated from Monte Carlo analysis.

4.2.2 CAS Distribution Data

This file provides distribution data to be analyzed by the post processor.

4.2.3 Segment Truth File

This file contains all of the essential Segment Truth data. It may be saved and input to LEM on a subsequent run allowing the Segment Truth Generator to be bypassed. However, on that subsequent run the Segment Truth data will be constant.

4.2.4 CAMS Output File

This file contains all of the essential CAMS output data. It may be saved and input to LEM on a subsequent run allowing CAMS to be bypassed. However, on that subsequent run the CAMS data will be constant.

4.2.5 YES Output File

This file contains all of the essential YES output data. It may be saved and input to LEM on a subsequent run allowing YES to be bypassed. However, on that subsequent run the YES data will be constant.

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many errors as possible during the processing of the input control cards. The program will continue checking for additional input errors if any input error is detected. There are two levels of error. These are:

Level 1 - non-fatal, continue processing.

Level 2 - job fatal. Terminate job after processing all input control cards.

When a level 1 error is detected, the program will print an informative message and continue processing. When a level 2 error is detected, the program will print an informative message, set a fatal error flag, and continue processing. When all control cards have been processed the program will continue executing if no fatal errors were found or will return control back to the operating system if at least one fatal error is detected.

The errors which may be detected by the LEM control program itself are described below. Any error conditions which are detected by CAMS, YES, or CAS will be described separately in the Problem Definitions for CAMS, YES, or CAS.

5.2 INPUT ERRORS DETECTED BY LEM

1. Message:

   TOO MANY MONTE CARLO TRIALS REQUESTED. NTRIAL = n, RSTART = r. MAX. NO. OF TRIALS PER RUN IS m.

   Meaning:

   On the LEM control cards the user has specified n - r Monte Carlo trials for the current run but the program permits a maximum of m trials for any single run.

   Remedy:

   Fatal error -- the user should check NTRIAL and RSTART and be sure NTRIAL-RSTART does not exceed the maximum allowable value.

2. Message:

RSTART = r MUST BE LESS THAN NTRIAL = n.

Meaning:

RSTART, the final iteration number from the previous run from which the user is trying to restart, must be less than NTRIAL, the total number of iterations desired at the end of the current run.

Remedy:

Fatal error -- RSTART is fixed. Hence, NTRIAL must be increased on the LEM control cards.

3. Message:

STARTR = n_1 MUST BE BETWEEN 0 AND ENDR = n_2. ENDR MUST BE .LE. m.

Meaning:

The starting region n_1 and the ending region n_2 must satisfy the inequalities

$$0 \leq n_1 \leq n_2 \leq m$$

where m is the maximum region number.

Remedy:

Fatal error -- the user should check STARTR and ENDR on the LEM control cards to be sure they satisfy the above inequalities.

4. Message:

STARTZ = n_1 MUST BE BETWEEN 0 AND ENDZ = n_2. ENDZ MUST BE .LE. m.

Meaning:

The starting zone n_1 and the ending zone n_2 must satisfy the inequalities

$$0 \leq n_1 \leq n_2 \leq m$$

where m is the maximum zone number.

Remedy:

Fatal error -- the user should check STARTZ and ENDZ on the LEM control cards to be sure they satisfy the above inequalities.

5. Message:

ISTG = n_1, ICAMS = n_2, AND IYES = n_3 MUST ALL BE 0, 1, 2, OR 3.

Meaning:

One or more of the parameters ISTG, ICAMS, and IYES have an illegal value specified. The only allowable values are 0, 1, 2, or 3.

Remedy:

Fatal error -- specify the proper value(s) for the offending parameter(s) on the LEM control cards.

6. Message:

IF ICAMS IS NONZERO, THEN ISTG MUST BE NONZERO. I.E. IF THE CAMS ERRORS ARE HELD CONSTANT, THEN SO MUST THE SEGMENT TRUTH ERROR.

Meaning:

Self-explanatory. The user cannot vary the Segment Truth error while holding the CAMS errors constant by setting ICAMS $\neq 0$. However, it is possible to hold the Segment Truth error constant while varying the CAMS errors.

Remedy:

Fatal error -- change either ICAMS or ISTG on the LEM control cards.

7. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM SEGMENT ID FILE DOES NOT AGREE WITH INPUTS ICSESG = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong Segment ID file has been specified, or the wrong values have been specified for the parameters ICSESG and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper Segment ID file or specify the correct values for ICSESG and CUNTRY on the LEM control cards. It might be necessary to dump the header record of the Segment ID file.

8. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM CROP WINDOW FILE DOES NOT AGREE WITH INPUTS ICSECW = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong Crop Window file has been specified, or the wrong values have been specified for the parameters ICSECW and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper Crop Window file or specify the correct values for ICSECW and CUNTRY on the LEM control cards. It might be necessary to dump the header record of the Crop Window file.

9. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM CAMS ERROR FILE DOES NOT AGREE WITH INPU TS ICSECE = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong CAMS error file has been specified, or the wrong values have been specified for the parameters ICSECE and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper CAMS error file or specify the correct values for ICSECE and CUNTRY on the LEM control cards. It might be necessary to dump the header record of the CAMS error file.

10. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM THE SIGNATURE EXTENSION FILE DOES NOT AGREE WITH INPU TS ICSESE = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong signature extension file has been specified, or the wrong values have been specified for the parameters ICSESE and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper signature extension file or specify the correct values for ICSESE and CUNTRY. It might be necessary to dump the header record of the signature extension file.

11. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM THE DATA ACQUISITION FILE DOES NOT AGREE WITH INPUTS ICSEAC = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong Data Acquisition file has been specified, or the wrong values have been specified for the parameters ICSEAC and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper Data Acquisition file or specify the correct values of ICSEAC and CUNTRY. It might be necessary to dump the header record of the Data Acquisition file.

12. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM THE YES ERROR MODEL FILE DOES NOT AGREE WITH INPUTS ICSEYM = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong YES Error Model file has been specified or the wrong values have been specified for the parameters ICSEYM and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper YES Error Model file or specify the correct values for ICSEYM and CUNTRY. It might be necessary to dump the header record of the YES Error Model file.

13. Message:

CASE NUMBER = n_1 OR COUNTRY C_1 FROM THE SUBSTRATA HISTORICAL FILE DOES NOT AGREE WITH INPUTS ICSESH = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong Substrata Historical file has been specified or the wrong values have been specified for the parameters ICSESH and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper Substrata Historical file or specify the correct values for ICSESH and CUNTRY. It might be necessary to dump the header record of the Substrata Historical file.

14. Message:

CASE NUMBER = n_1 OR COUNTRY = C_1 FROM THE SEGMENT TRUTH FILE DOES NOT AGREE WITH INPUTS ICSEST = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong file has been specified as the Segment Truth file, or the wrong values have been specified for ICSEST and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper Segment Truth file or specify the correct values for ICSEST and CUNTRY. It might be necessary to dump the header record of the Segment Truth file.

15. Message:

CASE NUMBER = n_1 OR COUNTRY = C_1 FROM THE CAMS OUTPUT FILE DOES NOT AGREE WITH INPUTS ICSECO = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong file has been specified as the CAMS Output file, or the wrong values have been specified for ICSECO and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper CAMS Output file or specify the correct values for ICSECO and CUNTRY. It might be necessary to dump the header record of the CAMS Output file.

16. Message:

CASE NUMBER = n_1 OR COUNTRY = C_1 FROM THE YES OUTPUT FILE DOES NOT AGREE WITH INPUTS ICSEYS = n_2 AND CUNTRY = C_2.

Meaning:

Possibly the wrong file has been specified as the YES Output file, or the wrong values have been specified for ICSEYS and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper YES Output file or specify the correct values for ICSEYS and CUNTRY. It might be necessary to dump the header record of the YES Output file.

17. Message:

CASE NUMBER = n_1 OR COUNTRY = C_1 FROM THE CAS CUM OUTPUT FILE DOES NOT AGREE WITH INPUTS ICASE = n_2 AND CUNTRY = C_2.

Meaning:

On a restart run the case number and country of the CAS Cum Output file must agree with the parameters ICASE and CUNTRY on the LEM control cards. Possibly the wrong file has been mounted, or the wrong values have been specified for ICASE and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper CAMS Cum file or specify the proper values for ICASE and CUNTRY. It might be necessary to dump the header record of the CAMS Cum file.

18. Message:

CASE NUMBER = n_1 OR COUNTRY = C_1 FROM THE CAS DIST OUTPUT FILE DOES NOT AGREE WITH INPUTS ICASE = n_2 AND CUNTRY = C_2.

Meaning:

On a restart run the case number and country of the CAS Distribution Output file must agree with the parameters ICASE and CUNTRY on the LEM control cards. Possibly the wrong file has been mounted, or the wrong values have been specified for ICASE and CUNTRY on the LEM control cards.

Remedy:

Fatal error -- mount the proper CAS Distribution file or specify the proper values for ICASE and CUNTRY. It might be necessary to dump the header record of the CAS Distribution file.

19. Message:

IMPROPER HEADER LABEL ON FILE filename.
LABEL = label.

Meaning:

The file label specified in the first two words of the header record of the file "filename" does not correspond to the expected label. Possibly the wrong file has been mounted.

Remedy:

Mount the proper file. It might be necessary to dump the header record of the designated file.

20. Message:

IMPROPER LABEL AND SEQUENCE NUMBER ON A LEM CONTROL CARD. LABEL AND SEQ. NO. = ______.

Meaning:

The four LEM control cards are supposed to have LEM 1 0i entered in Columns 75-80 (where i = 1, 2, 3, or 4). Possibly the control cards are out of order.

Remedy:

Be sure the LEM control cards are in the proper order and that the label and sequence numbers are entered properly.

21. Message:

ITERATION NUMBER NT FROM filename FILE = n DOES NOT AGREE WITH RSTART = r FROM LEM CONTROL CARD.
where filename = CASCUM or CASDIS

Meaning:

The iteration number NT from both the CAS Cumulative file (CASCUM) and the CAS Distribution file (CASDIS) must be consistent with the value of RSTART specified on the LEM control cards. Possibly the wrong file has been specified or else RSTART is specified incorrectly.

Remedy:

Fatal error -- mount the proper file or specify the correct value for RSTART.

5.3 PROCESSING ERRORS

Each of the modules STG, CAMS, YES, and CAS performs its own error checking during execution. The error messages for CAMS and CAS are described separately in the problem descriptions for those modules. The error messages for the Segment Truth Generator and the YES module are described below.

5.3.1 Processing Errors Detected by the Segment Truth Generator

1. Message:

THE SEGMENT ID FILE AND THE SUBSTRATA HISTORICAL FILE ARE INCONSISTENT.

	SEGID	SUBHST
REGION	r1	r2
ZONE	z1	z2
STRATA	s1	s2
SUBSTRATA	k1	k2

Meaning:

The region, zone, strata, and/or substrata ID's do not agree between the SEGID file and the SUBHST file. Perhaps the wrong file has been specified for one or both files. This error message was intended primarily for checkout purposes. The error should not occur during production usage.

Remedy:

Non-fatal error -- the segment from the SEGID file will be dropped and execution will continue. However, the user should check both files to be sure the proper files have been specified. It may be necessary to dump part or all of one or both files.

2. Message:

SEGMENT σ IS NOT IN IDSEG FROM SUBHST FOR REGION r, ZONE z, STRATA s, SUBSTRATA k
SEGMENT WILL BE DROPPED.

Meaning:

The indicated segment ID σ from the Segment ID file was not found in the array IDSEG read from the SUBHST file. Apparently the SEGID file and the SUBHST file are inconsistent.

Remedy:

Non-fatal error -- the indicated segment will be ignored and execution will proceed. However, the user should check both files to be sure the proper files have been specified.

3. Message:

ERROR RETURN FROM BETAD ROUTINE.
IER = f . PWK = PW_k SIGMA = σ PWKI = PW_i

Meaning:

An error return from the Beta Distribution routine has occurred in STG. (See the writeup of BETAD for details.) The error flag f indicates the nature of the error.

f = 1 XBAR = PWK ($\overline{X} = PW_k$) is not in the range $0 \le \overline{X} \le 1$.
$\overline{X}$ was reset within BETAD.

f = 2 σ not in the range $0 \le \sigma \le \overline{X} \sqrt{\frac{1 - \overline{X}}{\overline{X} + \epsilon}}$
where $\epsilon = 10^{-4}$
σ was reset within BETAD.

f = 3 Fatal error
PW_{xi} could not be found within 35 iterations.

Remedy:

If IER = 1 or 2, the error is non-fatal and the program continues execution.

If IER = 3, then the user better check the values PW_k or CV_2 from the SUBHST file.

4. Message:

ERROR RETURN FROM BETAD ROUTINE.
IER = f . PMMEAN = PM_k SIGMA = σ

Meaning:

Similar to error message 3 described above except that here
XBAR = PMMEAN
($\overline{X} = PM_k$)

Remedy:

If IER = 1 or 2, the error is non-fatal and the program continues execution.

If IER = 3, then the user should check the values PW_k, DELTPM, and CV_3 from the SUBHST file.

5. Message:

WARNING... NSEGS = n .NE. NSEG
(FROM SUBHST) = m
SEGMENT IDS MAY BE INCORRECT.

Meaning:

The number of segments read from the SEGID file does not agree with NSEG, the number of segments on the SUBHST file. Possibly the two files are inconsistent. The error is considered to be non-fatal by the program, but the error may be quite serious and the results should be regarded with suspicion.

Remedy:

Check to be sure the SEGID and SUBHST files are consistent. It might be necessary to dump all or part of both files.

6. Message:

WARNING... NO SEGMENTS PROCESSED BY SEGMENT TRUTH GENERATOR.

Meaning:

For some reason the Segment Truth Generator failed to process any segments. This is a very degenerate situation and should not normally be encountered in production runs. The most likely explanation is that all of the substrata processed by STG had zero segments.

Remedy:

Non-fatal error -- however, this condition will probably cause an abort in CAMS. Check the SUBHS file. Also check STARTR, STARTZ, ENDR, ENDZ.

PART I

PROBLEM DESCRIPTION

FOR THE CAMS SUBPROGRAM

Table of Contents

Problem Description - CAMS

1.0 SCOPE

1.1 PROBLEM CAPABILITIES

The CAMS model provides an estimate of the proportion wheat in each segment selected by the acquisition program. Up to four estimates per segment are given, one for an acquisition date in each of four crop windows. It gives a Monte Carlo simulation of three types of errors:

1. Crop calendar errors
2. Signature extension errors (ordinary segments)
3. Classification errors - an error category including crop calendar, multi-temporal sampling effects, and input classification errors (training segments)

A fourth type of error modeled is multi-temporal sampling effects, which remains constant during a run. A fifth type of error is allowed by the choice between two classification models. The more complex model 1 allows for mixed crops, and includes the effect of omission and commission errors - the confusion created by other crops growing in the area. Options allow the omission of the effects of each type of error.

Since the signature extension error is not well understood, CAMS allows for the choice of either an additive or multiplicative factor. Signature extension requires acquisition of a training segment within a fixed period preceding the ordinary segment acquisition. If this condition is not met, then the ordinary segment is either classified as a training segment or not classified at all. The option desired is controlled by user input.

CAMS requires five input files and some card input. It produces an output file for CAS to use for aggregation of the wheat area estimates. On option, it also produces a report of the wheat estimates, and on option also, a breakdown of the error factors.

1.2 PROGRAM DEVELOPMENT AND ORGANIZATION

CAMS will be developed in FORTRAN as an overlay of the LEM program. See the LEM problem description, Section 1.2.

1.3 OPERATIONAL ASSUMPTIONS

See the LEM problem description, Section 1.3.

2.0 INPUT

CAMS requires five input files and some card data.

2.1 CARDS

Some data needed by CAMS is included on the LEM control card See LEM problem description, Section 2.1. CAMS also requires:

1. A control card, specifying options
2. A multi-temporal matrix
3. Crop calendar error coefficients

Besides the LEM card, CAMS requires a total of 13 cards, which must be in order. See Figure 4 for the deck setup.

2.1.1 List of Data Quantities and Formats

a. LEM control card, see LEM Section 2.1. Data relevant to CAMS includes:

ISEXT	Signature extension error option, = 0, 1 simulate error = 2 bypass error
ISCC	Crop calendar error option, = 0, 1 simulate = 2 bypass
ICLASS	Classification error option, = 0, 1 simulate = 2 bypass
IACQ	Acquisition file option, = 0 include file = 1 no acquisition file
ICAMS	CAMS error option, = 0, 1 simulate = 3 bypass all errors
SEED2	Random no. seed for classification error
SEED3	Random no. seed for signature extension error
SEED4	Random no. seed for crop calendar error
IPRCAM	Print flag for CAMS - this controls if a report is printed - the flag, IREP, on the CAMS control card, controls what is printed.

b. CAMS control card, see Figure 1 for the format and list of quantities. See Figure 4 for the total deck setup.

c. Multi-temporal sampling matrix. The multi-temporal sampling model describes the effect of the acquisition of a sample segment in more than one bio-window. There are 15 possible non-zero acquisition states for a sample segment. These states are shown below.

Windows included	1	2	3	4	1, 2	1, 3	1, 4	2, 3	2, 4	3, 4	1, 2, 3	1, 2, 4	1, 3, 4	2, 3, 4	1, 2, 3, 4
Group #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

The effect of acquisition conditions corresponding to any particular state is modeled by a weighting factor, M. For ease of handling, three values of M are chosen as program inputs associated with groupings of the 15 acquisition states. These values indicate no improvement ($M_1 = 1$), small improvement ($M_2 < 1$), and large improvement ($M_3 < M_2$) in the accuracy of classification of the segment with a given acquisition state over classification in the present window (last window in the state definition) alone.

The values needed are which M (M1, M2, or M3) to use for each of the 15 groupings, and the values of M2 and M3 ($M1 \equiv 1$). Note that by definition, for group no. 1, 2, 3, and 4, M1 = 1, and M1 = 1 > M2 > M3. See Figure 2 for the description.

A total of eight cards are needed for the matrix, each with the above information, since the acquisition conditions depend on a. wheat type - winter or spring, b. the model - 1 or 2 (model 1 requires three cards, one for wheat, mixed, and other components; model 2 requires only one card). See Figure 4 for the order and setup.

If model 1 is being used, the cards for model 2 must be present but may have blank fields except for the id (CAMS) and sequence number, and vice versa if model 2 is being used, since the values are not used. If data for both winter and spring is not available, the data may be left blank (except id and sequence number), but the cards must be present, and if CAMS tries to use the missing data, an error will be reported. If the multi-temporal error bypass is specified (IMULTI=1), then all the cards must be present but all the data but the id and sequence number may be left blank.

d. Crop calendar coefficients. The effect of crop calendar errors on segment classification, particularly in an analytic sense, is not well established at this time. For this reason, a simple generic model was chosen to represent this effect. This model generates a bias (B) and standard deviation (σ) from a quadratic function with user input coefficients.

$$B = G_1 (\Delta t) + G_2 (\Delta t)^2$$

$$\sigma = H_1 (\Delta t) + H_2 (\Delta t)^2$$

The value of the coefficients are to be determined from off-line analysis, curve fitting, etc., to represent the observed effects.

The values for G1, G2, H1, and H2 are needed. See Figure 3 for a description of the quantities and format. Model 1 requires these four values for the three components, wheat, mixed, and other, a total of 12 values. Model 2 requires only the four values, since the mixed crop effect is not present. Since these values may be different for winter and spring wheat, two sets must be inputted. Thus, four cards are always needed:

1. Spring wheat - model 1 - 3x4 values
2. Spring wheat - model 2 - 4 values
3. Winter wheat - model 1 - 3x4 values
4. Winter wheat - model 2 - 4 values

If model 1 is used, the cards for model 2 must be present but may have blank fields except for the id, CAMS, and sequence number, and vice versa for model 2. If data for both winter and spring is not available, the data may be left blank except for id and sequence number, but the cards must be present, and if CAMS needs the missing data, an error will be reported. If the crop calendar error bypass is specified (ISCC=2), then all the cards must still be present, but all the id and sequence number fields may be left blank. See Figure 4 for the full CAMS deck setup.

2.1.2 Deck Setup

See Figure 4. CAMS requires 13 card inputs.

2.1.4 Rules for Entering Data

See LEM problem description, Section 2.1.4, for general rules for entering data.

2.2 FILES

CAMS requires five input files:

1.	CAMS error file (CAMERR)	from SEE program
2.	Segment truth file (SEGTRU)	intermediate LEM file
3.	Data acquisition file (ACQUIS)	from SACS program
4.	Crop calendar file (CROPW)	from LUMP program
5.	Signature extension file (SIGEXT)	from SEE program

See the file description, Section 2.4 of the Users Manual, for the formats and contents. If certain error simulation types are bypassed, the associated input file need not be loaded. See Section 3.3 for a description of all the CAMS error control flags and error bypasses. If the IACQ flag is 1 on the LEM control card, all CAMS error simulation is bypassed, and only the SEGTRU and CROPW files need be loaded. If the crop calendar error is bypassed (ISCC=2), the CROPW flag file need not be loaded. If the signature extension error is bypassed (ISEXT=2), the SIGEXT file need not be loaded. The ICAMS and ICLASS flags can cause more than one type error to be bypassed and the appropriate associated input files need not be loaded.

Name	Dimension	Nominal Value	Range	Units	Description
IMODEL			1-2		Flag = 1 use model 1, complex model = 2 use model 2, simple model
IMULTI		0	0-1		Flag = 0 include multi-temporal sampling error ≠ 0 bypass multi-temporal sampling error
ISIGEX		0	0-1		Flag ≠ 0 use multiplicative model of signature extension = 0 use additive model of signature extension
ISKIP		0	0-1		Flag = 0 skip; ≠ 0 classify as training — if cannot correlate ordinary with training segment
ITMAX			0-99	Days	Maximum no. of days between training and ordinary segment acquisition dates for successful correlation.
IREP		0	0-1		Flag = 0 include error breakdown factors in estimate report ≠ 0 print estimate report only
IWIND		4	1-4		From which window to take the probability of classifying as wheat given mixed to calculate the proportion of <u>pure</u> wheat pixels; if blank, defaults to window 4. This quantity is $\overline{P}(W/M)$ on the CAMERR input file; see file descriptions, Section

Input Data - CAMS Control Card Format

c. c.	1	2	3	4	5	7	8
	IMODEL	IMULTI	ISIGEX	ISKIP	ITMAX	IREP	IWIND
	I1	I1	I1	I1	I2	I1	I1

75	79
C A M S	0 1
A4	I2

Figure 1. Input Data - CAMS Control Card Quantities

Name	Dimension	Nominal Value	Range	Units	Description
IGROUP	15		1-3		Which value of M to use for each acquisition state, see 2.1.1, c., for which windows are in each state = 1 use M1 (≡ 1), no improvement = 2 use M2, small improvement = 3 use M3, large improvement Restriction: IGROUP(1), IGROUP(2), IGROUP(3) and IGROUP(4) are always = 1, by definition, and so need not be inputted.
M2			0.0<M2<1.0		Value of M2, small improvement
M3			0.0<M3<1.0		Value of M3, large improvement Restriction: M3 < M2 < 1, by definition.
ISEQ			2-9		Sequence no. - the matrix requires eight cards, each with the same format. See Deck-Setup, Figure 4, for the order and description.

Input Data - Multi-Temporal Matrix Format

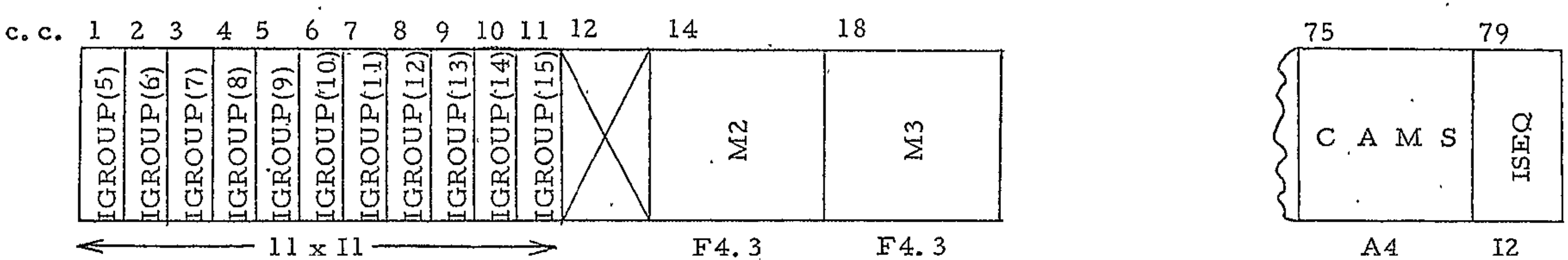
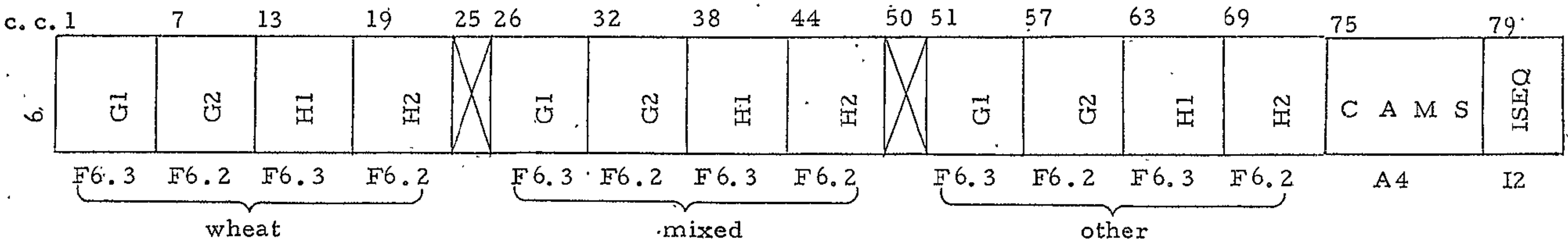


Figure 2. Input Data - Multi-Temporal Matrix Quantities

Name	Dimension	Nominal Value	Range	Units	Description
G1			±9.999		Crop calendar error coefficient
G2			±99.99		Crop calendar error coefficient
H1			±9.999		Crop calendar error coefficient
H2			±99.99		Crop calendar error coefficient
ISEQ			10-13		Sequence number - the calendar requires four cards. See Deck Setup, Figure 4.

Input Data - Crop Calendar Format

Model 1:



Model 2:

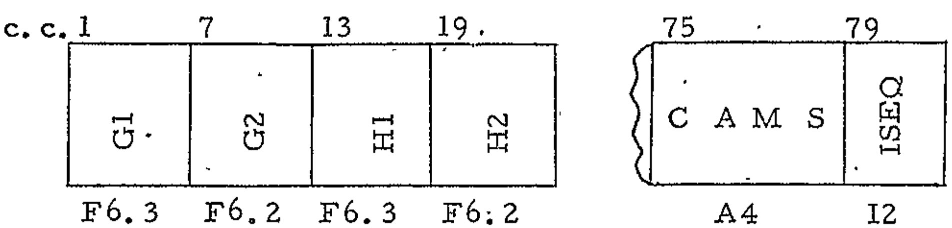


Figure 3. Input Data - Crop Calendar Coefficients Quantities

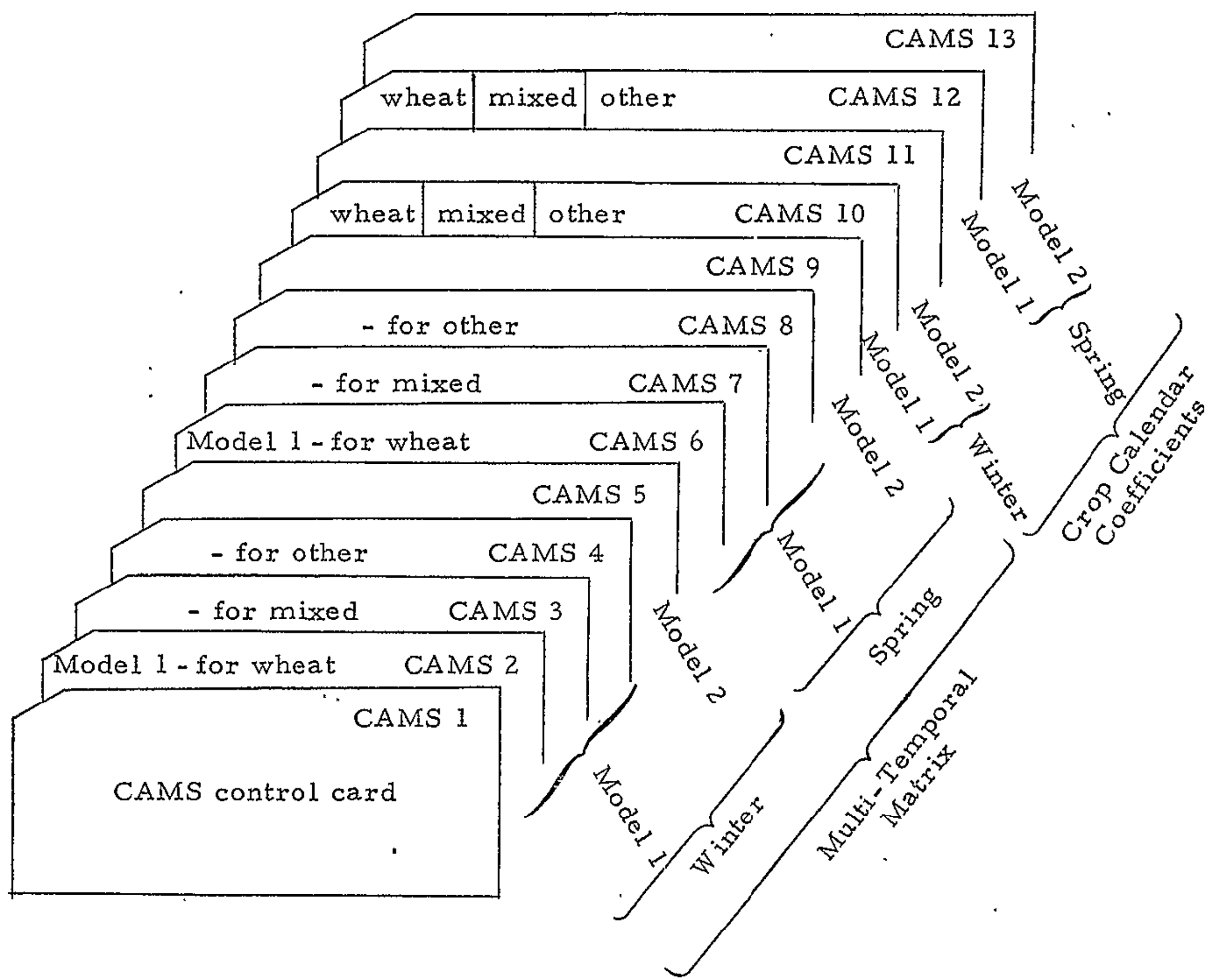


Figure.4. CAMS Deck Setup

3.0 PROCESSING

3.1 OVERVIEW

See Figure 5 for an overview of the program flow in CAMS. CAMS operates on two passes through the acquisition data. The training segments are classified on the first pass and the ordinary segments on the second pass. The flow for models 1 and 2 are essentially the same, but for the more complex model 1 the error factors must be computed for each of the three divisions, wheat, mixed, and other, whereas for model 2 this breakdown does not take place. A random access scratch file with all the training segments is needed to figure out correlation of ordinary with training segments.

3.2 PROGRAM FLOW

See Figure 6 for a block diagram of CAMS, at a more detailed level than Figure 5. It shows the possible options allowed by CAMS through its control card. The subroutine CAMSIN, shown in Figure 5, is called by LEM to read in the CAMS control card, multi-temporal matrix, and crop calendar coefficients, which are passed to the actual CAMS subprogram by LEM after error checking.

3.3 PROCEDURES AND EQUATIONS

This corresponds to the Figure 6 flowchart. It specifies all the necessary equations.

CAMS flow can be broken down into:

A. Initialization

B. Pass 1 - training segments

   1. Compute multi-temporal effects
   2. Compute crop calendar errors
   3. Compute input classification and total classification error (includes 1 and 2)
   4. Compute PEST, estimated proportion wheat
   5. Write report

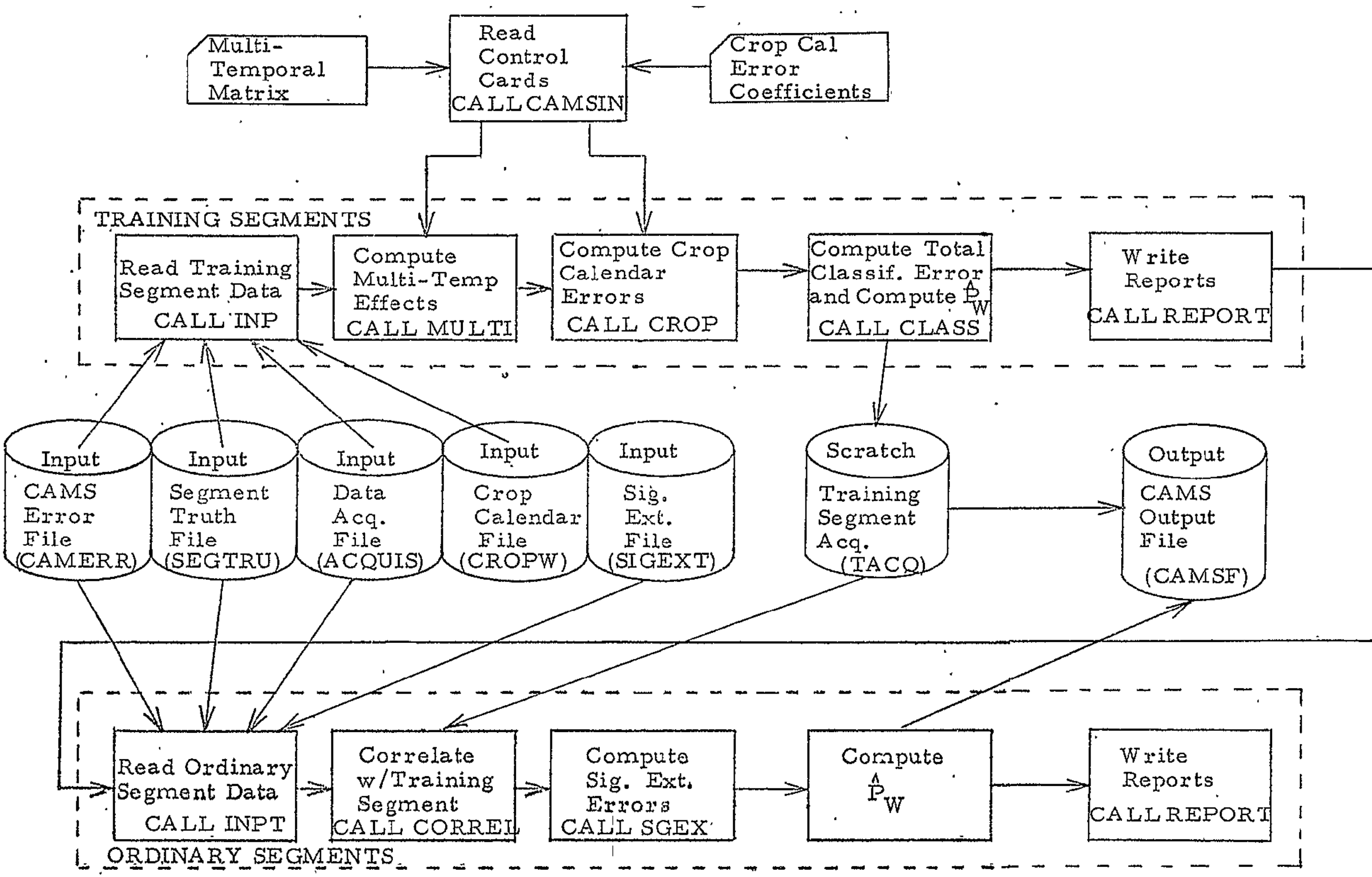


Figure 5. FOC CAMS Error Model

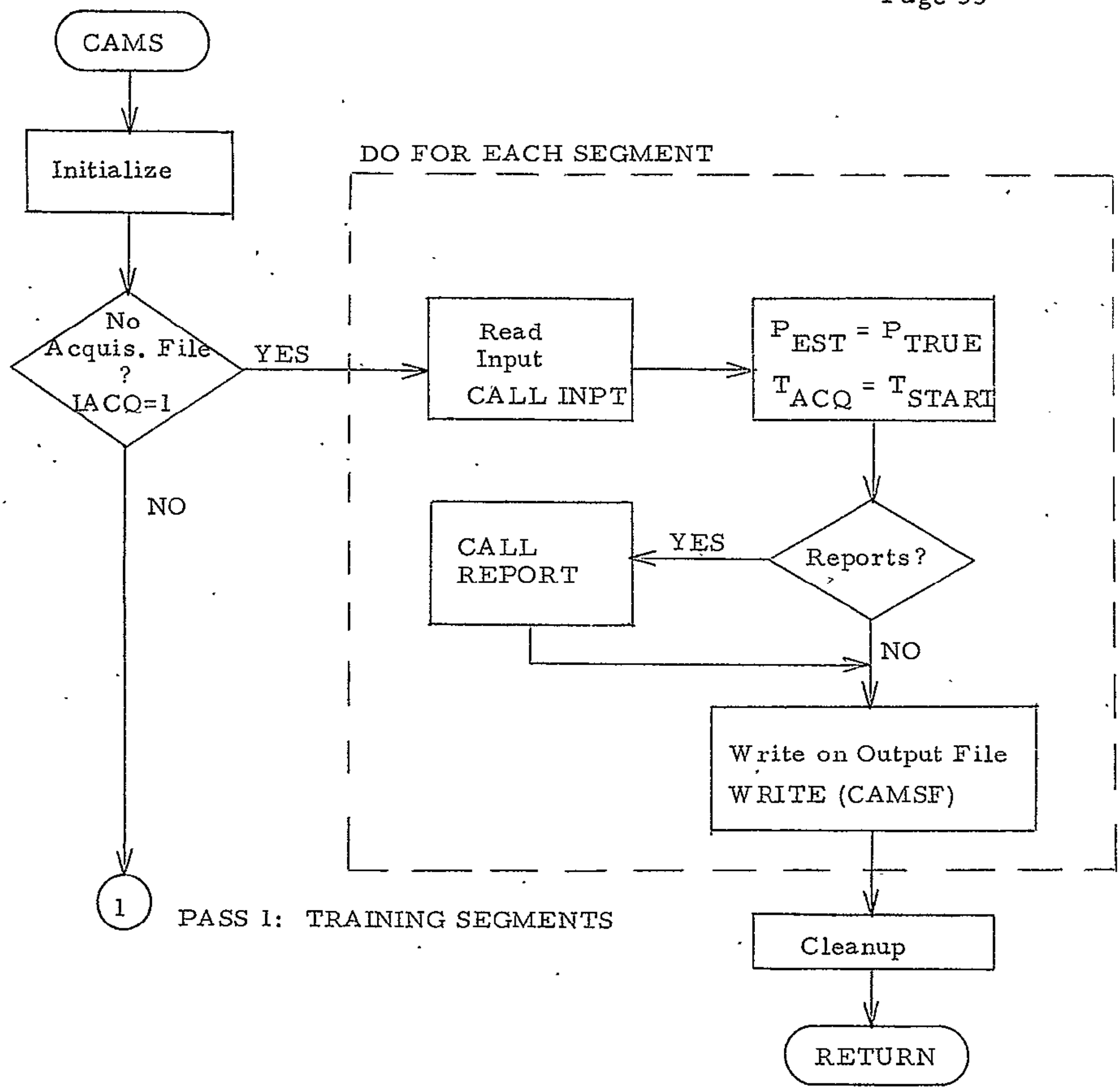


Figure 6. CAMS Logic Flow (Sheet 1 of 4)

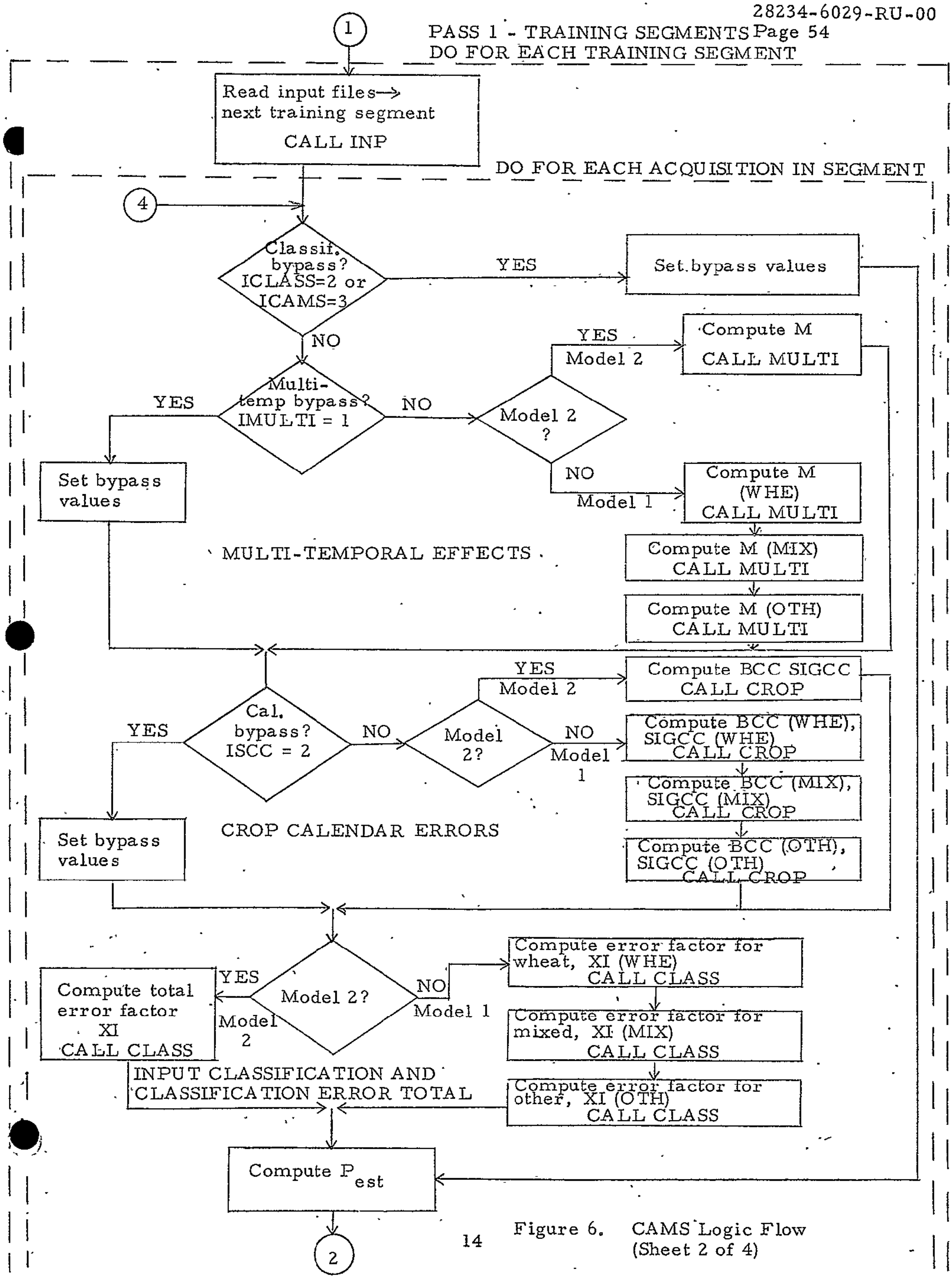


Figure 6. CAMS Logic Flow
(Sheet 2 of 4)

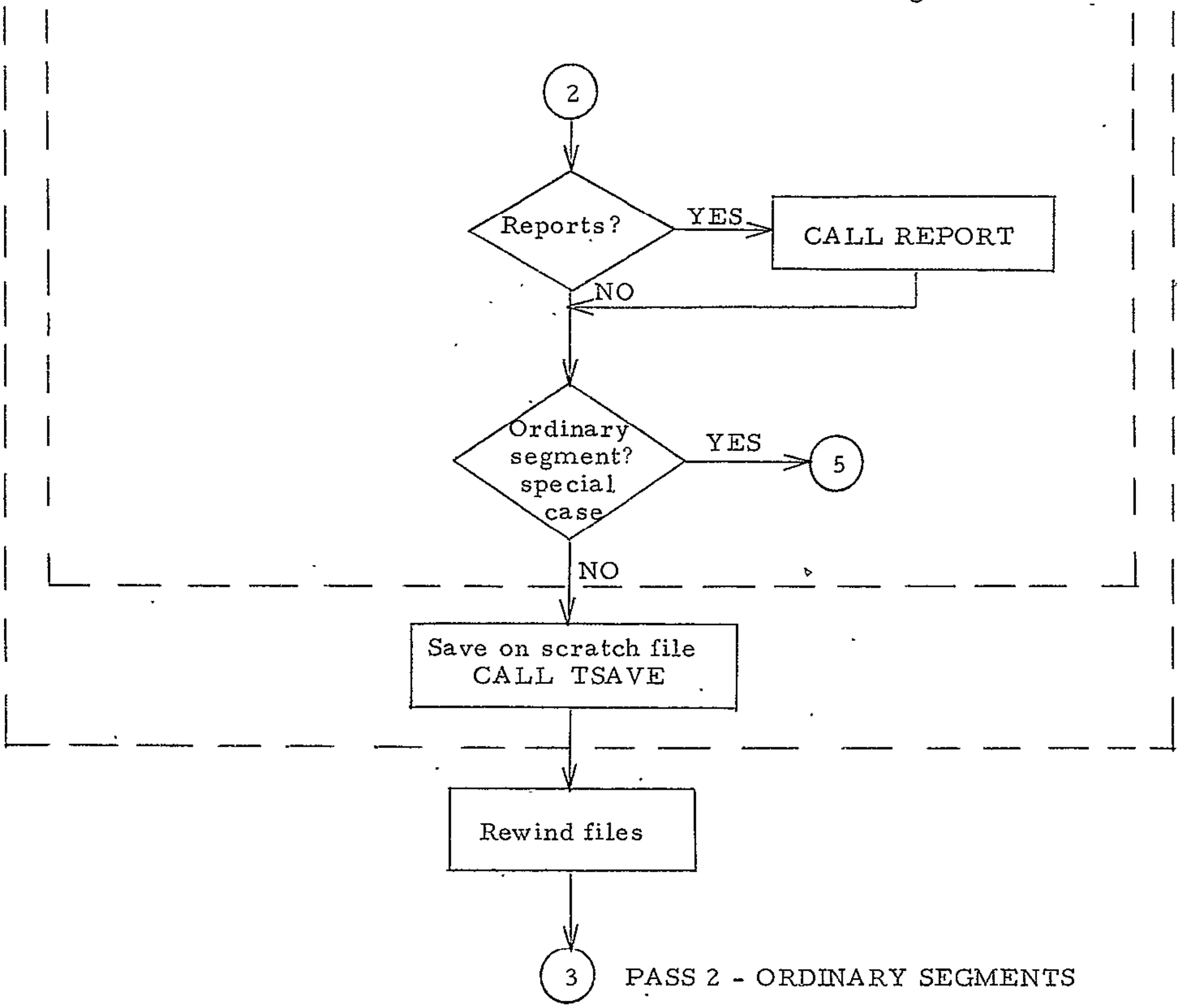


Figure 6. CAMS Logic Flow (Sheet 3 of 4)

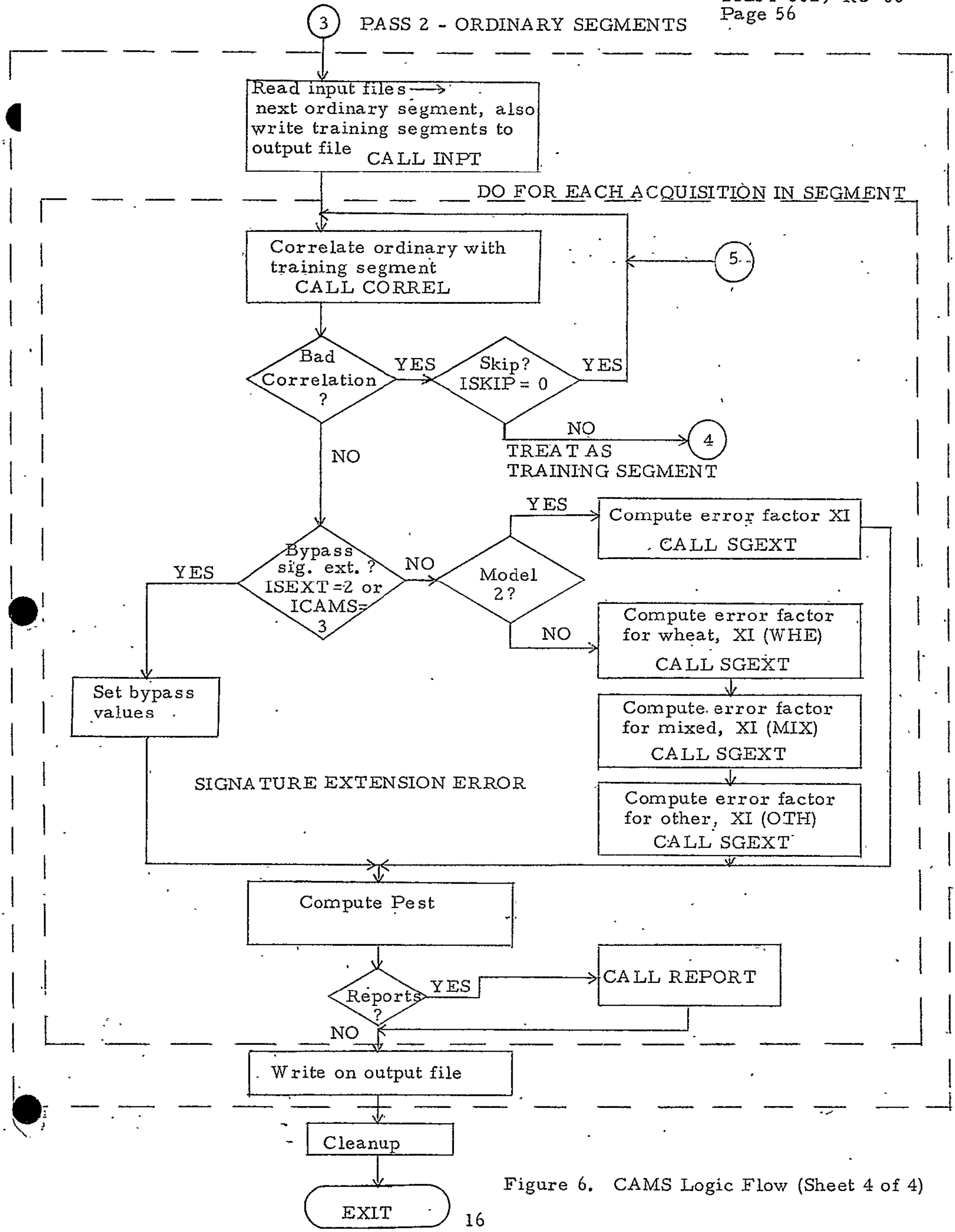


Figure 6. CAMS Logic Flow (Sheet 4 of 4)

C. Pass 2 - ordinary segments

1. Correlate with training segment

2. Compute signature extension error (includes B.3. for associated training segment)

3. Compute PEST, estimated proportion wheat

4. Write report

This is the flow of Figure 5. Figure 6 includes the complications introduced by options to bypass error effects, summarized below:

Error Type: Flag:	IMULTI	ISCC	ICLASS	ISEXT	ICAMS	IMODEL	IACQ
1. Multi-temporal	X		X		X		X
2. Crop calendar		X	X		X		X
3. Input classification			X		X		X
4. Signature extension				X	X		X
5. Mixed crops						X	X

where X means error effect is bypassed.

If a more general flag is on, it will overrule the more specific flags.

All flags are on either the LEM or CAMS control cards. The IMODEL variable is not a flag, but specifies the model 1 or 2, but specifying model 2 has the effect of bypassing the mixed crop effects. What these mean, applied to the equations, will be spelled out later. In the equations, the W, M, and O stand for terms associated with wheat, mixed, and other. For model 2, M and O terms are zero. Note that the equations are presented from final to start.

B. Pass 1 - training segment acquisitions

B1. Compute PEST

The heart of this pass is the calculation of PEST, the estimated proportion of wheat:

1a. PEST = P(W) * XI(W) + P(M) * XI(M) + P(O) * XI(O)

where P(W), P(M), and P(O) true proportion of pure wheat, mixed, and pure other pixels

The total error would then be:

1b. PERR = PEST - PT(W)

1c. P(W) = PT(W) - PT(M) * PW(M, IWIND)

PT(W), PT(M) from SEGTRU file

Note: If out of range, PT(M) recomputed as:

If PT(M) * PW(M, IWIND) < PT(M) + PT(W) - 100

then PT(M) = (100 - PT(W))/(100 - PW(M, IWIND)

If PT(M) * PW(M, IWIND) > PT(W)

then PT(M) = PT(W)/PW(M, IWIND)

PW(M, IWIND) from CAMERR file

IWIND from CAMS control card

1d. P(M) = PT(M)

1e. P(O) = 100 - P(M) - P(W)

and where XI(W), XI(M), and XI(O) are the probability of classifying as wheat, given wheat, mixed, or other, and includes all the error factors

B2. Compute XI (type) - classification error

The XIs are computed by first computing XBARS and SIGMAS and then getting a random number from a Beta distribution.

2a. CALL BETAD (SEED(2), XBAR (TYPE), SIGMA (TYPE), XI (TYPE), O)

where SEED(2) is the random no. seed from card input for classification error

TYPE = W, M, O - call Betad three times

2b. SIGMA (TYPE) = PW (TYPE, WINDOW) *M (TYPE) *SIG (TYPE)

where PW from CAMERR file

M multi-temporal error factor

SIG crop calendar/input classification error factor sigma

WINDOW which window current acquisition date in = 1, 2, 3, or 4

2c. XBAR (TYPE) = PW (TYPE, WINDOW) * (1. + M (TYPE) * B (TYPE))

where PW same as above

M same as above

B crop calendar/input classification error factor bias

Add input classification errors

3a. B (TYPE) = BERR (TYPE, WINDOW) + BCC (TYPE)

where		
	BERR	input classification error bias from CAMERR file
	BCC	crop calendar error bias

3b. SIG (TYPE) =

SQRT (SIGERR (TYPE, WINDOW) * SIGERR (TYPE, WINDOW) + SIGCC (TYPE) * SIGCC (TYPE))

where		
	SIGERR	input classification error sigma from CAMERR
	SIGCC	crop calendar error sigma

B3. Compute BCC (TYPE), SIGCC (TYPE) - crop calendar error.

Compute the crop calendar error, BCC and SIGCC:

4a. BCC = G (TYPE, SEASON, 1) * DELTA +
G (TYPE, SEASON, 2) * DELTA * DELTA

where		
	G	from card input
	SEASON	winter or spring, from SEGTRU
	DELTA	difference between true and observed windows

4b. SIGCC = ABS (H (TYPE, SEASON, 1) * DELTA +
H (TYPE, SEASON, 2) * DELTA * DELTA)

where		
	H	from card input
	SEASON	same as above
	DELTA	same as above

4c. DELTA = (TSEG - TSTART) / (ENDSEG - BGNSEG + 1)

4d. TSTART = ERR (SEASON, WINDOW)

ERR from CROPW file

4e. ENDSEG = END (SEASON, WINDOW) + TSEG

BGNSEG = START (SEASON, WINDOW) + TSEG

END, START from CROPW file

Compute TSEG only for first acquisition in each window; for rest, use same value.

5a. ITSEG = RN*SD (SEASON)

SD from CROPW file

5b. CALL BETAD (SEED(4), 0, 0, RN, 1, IER)

SEED(4) crop calendar seed, from card input returns

RN random no. from normal distribution

B4. Compute M (TYPE). - multi-temporal matrix error factor. This is just a table lookup.

6a. M (TYPE) = MS (TYPE, SEASON, IWHATM)

MS from card input

6b. IWHATM = IGROUP (TYPE, SEASON, ISTATE)

IGROUP from card input

6c. ISTATE = INDEX (IWIN(1), IWIN(2), IWIN(3), IWIN(4))

INDEX local array set up to yield correct value

IWIN local array calculated in CAMS specifying which windows have had acquisitions processed, = 1 no, = 2 yes

These are all the equations necessary to compute PEST. If model 2 is specified, TYPE=WHEAT only for all steps is computed, PW (W, WINDOW)=PT(W)/100, and also 1a. becomes P(W)=100. If the ICLASS or ICAMS bypasses are specified,

XI (TYPE) = PW (TYPE, WINDOW)

where PW from CAMERR file

and only step B1. needs to be done. If IMULTI bypass is specified, step B4. is skipped and M (TYPE) = 1. If ISCC bypass is specified, step B3. is skipped and BCC (TYPE) = , SIGCC (TYPE) = 0.

The acquisition date for each window on the output fi e is set to the first acquisition date in each window from the ACQUIS file.

Error factors must be calculated and saved for each error type for the error report, if necessary. These equations are:

For total error:

1c. TOT = PERR

1d. V (TYPE) = (XI (TYPE) - XBAR (TYPE))/ (PW (TYPE, WINDOW) * M (TYPE))

1e. ERTOT (TYPE) = M (TYPE) * (B (TYPE) + V (TYPE))

1f. ERBIAS (TYPE) = M (TYPE) * B (TYPE)

1g. ERRAND (TYPE) = M (TYPE) * V (TYPE)

For classification error:

2d. CLTOT (TYPE) = B (TYPE) + V (TYPE).

2e. CLBIAS (TYPE) = B (TYPE)

2f. CLRAND (TYPE) = V (TYPE)

For crop calendar:

5f. DELTA

5g. CROPD = TSEG - TSTART

For multi-temporal:

6d. MULT (TYPE) = M (TYPE)

C. Pass 2 - ordinary segments

C1. Compute PEST

The heart of this pass is the calculation of PEST, the estimated proportion of wheat. The equations are the same as for B. Pass 1, 1a-1d. However, the error factors XI are computed differently.

C2. Compute XI (TYPE) - signature extension error

The XI's are again computed from XBARs and SIGMAS, and picking a random no. from a Beta distribution.

2a. CALL BETAD (SEED(3), XBAR (TYPE), SIGMA (TYPE), XI (TYPE), O)

where	SEED(3)	is the random no. seed from card input for signature extension error

2b. XBAR (TYPE) = PW (TYPE, WINDOW) * (1. + TERTOT (TYPE) * ZB (TYPE, 1) + ZB (TYPE, 2))

where	PW	from CAMERR file
	TERTOT	from training segment value for ERTOT
	ZBs	from SIGEXT file ZB (TYPE, 1) = ZB (TYPE, 1) + 1 - multiplicative factor is increased by 1 from value on SIGEXT file

2c. SIGMA (TYPE) = PW (TYPE, WINDOW) *
ABS (TERTOT (TYPE) * ISIGEX * ZSIG (TYPE, 1, IUSE)
+ (1 - ISIGEX) *ZSIG (TYPE, 2, IUSE))

where	PW	from CAMERR file
	TERTOT	same as above
	ZSIG	from SIGEXT file
	ISIGEX	from card input
	IUSE	calculated during correlation

C3. Compute IUSE - correlation of training with ordinary segment

3a. IUSE = I of IPRIOR(I), I = 1, 6 for training segment which was ok; if none, I = 7

where IPRIOR from SEGTRU file

For each training segment until good one:

3b. Find closest training acquisition date previous to acquisition date of ordinary segment

3c. Subtract the two and check against
ITMAX = max. no. of days from card input

These are all the equations needed to compute PEST. The output acquisition dates are from the ACQUIS file. If model 2 is specified, TYPE = WHEAT only for all steps, PW(W., WINDOW) = PT(W)/100, and also 1a becomes P(W) = 100. If the ISEXT or ICAMS bypasses are specified, instead of B2 step, do:

XI (TYPE) = PW (TYPE, WINDOW) * (1 + TERTOT (TYPE))

where	PW	from CAMERR file
	TERTOT	from training segment value for ERTOT

If there was no correlation, the segment is either skipped or treated just like a training segment for that acquisition. If the IACQ bypass is specified, PEST = PT(W) and no error calculations for either training or ordinary segments are done. The acquisition date on the output file is set to START (WINDOW) from the CROPW file.

Error factors must be calculated and saved for each error type for the error report, if necessary. These equations are:

For total error:

1c. TOT = PERR

where PT(W) from SEGTRU file

1d. SE = PEST

1d. ALOCAL = P(W) * (1 + TERTOT(W)) * PW(W, WINDOW) +
P(M) * (1 + TERTOT(M)) * PW(M, WINDOW) +
P(O) * (1 + TERTOT(O)) * PW(O, WINDOW)

1e. ERTOT (TYPE) = TERTOT (TYPE) * Z (TYPE, 1) + Z (TYPE, 2)

1f. V (TYPE, 1) = (XI (TYPE) - XBAR (TYPE))/
(PW (TYPE, WINDOW) * TERTOT (TYPE))
V (TYPE, 2) = 0
} ISIGEXT = 1

V (TYPE, 1) = 0
V (TYPE, 2) = (XI (TYPE) - XBAR (TYPE)) /
PW (TYPE, WINDOW)
} ISIGEXT = 0

Note: If SIGMA=0, then V(TYPE, 2)=0, V(TYPE, 1)=0, and no calculation done.

1g. ERBIAS (TYPE) = TM (TYPE) * TB (TYPE) * ZB (TYPE, 1)
+ ZB (TYPE, 2)

where TM and TB are M and B of assoc. training segment

1h. ERRAND (TYPE) = TM (TYPE) *
(TV(TYPE) * ZB (TYPE, 1) + TB (TYPE) * V (TYPE, 1)
+ TV (TYPE) * V (TYPE, 1)) + V (TYPE, 2)

where TV is V from assoc. training segment

For classification error:

2d. CLTOT (TYPE) = (TB (TYPE) + TV (TYPE))
* Z(TYPE, 1) + Z(TYPE, 2)

2e. CLBIAS (TYPE) = TB (TYPE)*ZB(TYPE, 1)+ZB(TYPE, 2)

2f. CLRAND (TYPE) = TV (TYPE)*ZB(TYPE, 1)
+ TB (TYPE) * V (TYPE, 1) + TV (TYPE) * V (TYPE, 1)
+ V (TYPE, 2)

For signature extension:

2g. Z (TYPE, 1) = ZB (TYPE, 1) + V (TYPE, 1)

2h. Z (TYPE, 2) = ZB (TYPE, 2) + V (TYPE, 2)

For training segment:

2i. PID = IPRIOR (IUSE)

2j. TRAINA = (PEST/ALOCAL) * 100

2k. TRAIND = (PEST - ALOCAL)/ALOCAL * 100

Note: If ALOCAL = 0, TRAINA = ∞
If ALOCAL = 0 and PEST = 0, TRAINA = 100, TRAIND = 0.

Again, for model 2, TYPE = WHEAT only.

4.0 OUTPUT

4.1 PRINT DATA

4.1.1 Reports

On option, CAMS produces a yield estimate report which, on option, also includes a breakdown of the error factors. These options are controlled by the IPRCAM flag on the LEM control card and the IREP flag on the CAMS control card.

Figure 7 gives the layout of the report. The report is divided into two parts, for training and ordinary segments. Section 3 gives the equations needed for each category.

4.1.2 Echo Print Input Card Images

The 13 CAMS input cards are always echo printed. Defaulted or missing data will appear as zeros.

4.2 FILES

CAMS outputs one output file, the CAMS output file (CAMSF), to be used by CAS. See the file description, Section 2.4 of the Users Manual for the format and contents.

COUNTRY XXXX, REGION XX, ZONE XXX, STRATA XXX, SUBSTRATA XXXX, TRAINING SEGMENT XXXX
TRUE PROPORTION WHEAT = XXX.XX

1

CROP WINDOW	ACQ DATE MO/DY/YR	ESTIM. PROP.	TOTAL ERROR		WHEAT	ERROR MIXED	OTHER	CLASSIF. ERROR WHEAT	MIXED	OTHER	CROP CAL		MULTI-TEMP
XXXXXXXXXXXXXXX	XX/XX/XX	(PES)	(TOT)	TOT	(ERTOT)	(ERTOT)	(ERTOT)	(CLTOT)	(CLTOT)	(CLTOT)	(DELTA)	W	(MULT)
				BIAS	(ERBIAS)	(ERBIAS)	(ERBIAS)	(CLBIAS)	(CLBIAS)	(CLBIAS)	(CROPD)	M	(MULT)
				RAND	(ERRAND)	(ERRAND)	(ERRAND)	(CLRAND)	(CLRAND)	(CLRAND)		O	(MULT)
	XX/XX/XX	XXX.XX	XX.XXX		X.XXXX	X.XXXX	X.XXXX	X.XXXX	X.XXXX	X.XXXX	.XX		.XX

⋮

* * * * * * * * * * * * * * * * * *   *

⋮

If only estimate report wanted, from 1 to the right omitted.

COUNTRY XXXX, REGION XX, ZONE XXX, STRATA XXX, SUBSTRATA XXXX, ORDINARY SEGMENT XXXX
TRUE PROPORTION WHEAT = XXX.XX

1

CROP WINDOW	ACQ DATE MO/DY/YR	ESTIM. PROP.	TOTAL ERROR		WHEAT	ERROR MIXED	OTHER	WHEAT	SIG. EXT. MIXED	OTHER		Z1	Z2	TRAIN SEG.
XXXXXXXXXXXXXXX	XX/XX/XX	(PES)	(TOT)	TOT	(ERTOT)	(ERTOT)	(ERTOT)	(CLTOT)	(CLTOT)	(CLTOT)	W	(Z)	(Z)	(TID)
				BIAS	(ERBIAS)	(ERBIAS)	(ERBIAS)	(CLBIAS)	(CLBIAS)	(CLBIAS)	M	(Z)	(Z)	(TRAINA)
			(ALOCAL)	RAND	(ERRAND)	(ERRAND)	(ERRAND)	(CLRAND)	(CLRAND)	(CLRAND)	O	(Z)	(Z)	(TRAIND)

Note: If model 2, MIXED and OTHER quantities will appear as zeros.
If unable to correlate ordinary segment acquisition with training segment and so treated as training segment, TRAIN SEG. column will appear as out of range (all *'s) and SIG. EXT. Z1 will have the CROP CAL data, and SIG. EXT. Z2 the MULTI-TEMP data.

Figure 7. CAMS Estimate and Error Reports

5.0 ERROR PROCESSING

5.1 GENERAL

See the LEM problem description, Section 5.1, for a summary of overall error handling.

5.2 INPUT DATA ERRORS

1. CAMS XX MODEL NOT 1 OR 2 - X

   Fatal error - model number on CAMS control card, CAMS 01, is not 1 or 2. See Figure 1.

2. CAMS XX ITMAX NOT BETWEEN 0 AND 99 - XX

   Fatal error - ITMAX value on CAMS control card bad. See Figure 1.

3. CAMS XX IWIND NOT BETWEEN 0 AND 4 - X

   Fatal error - IWIND value bad (note that 1-4 good values, 0 = default value, set to 4) on CAMS control card. See Figure 1.

4. BAD CAMS ID OR SEQUENCE NO. - XXX XX

   Fatal error - CAMS control cards bad, perhaps out of order, or missing one. See Figure 4.

5. CAMS XX CROP CALENDAR COEF. OUT OF RANGE - XXXXX

   Fatal error - crop calendar coefficient should be between ± 9.999 or ± 99.99. See Figure 3.

6. CAMS XX BAD MULTI-TEMPORAL MATRIX VALUE M(X) - XXXX

   Fatal error - M2 not in range $M3 \leq M2 < 1.0$ or M3 not in range $0 \leq M3 \leq M2$. See Figure 2.

7. CAMS XX BAD MULTI-TEMPORAL MATRIX VALUE IGROUP (XX) - X

   Fatal error - IGROUP value not 1, 2, or 3. See Figure 2.

5.3 PROCESSING ERRORS

1. CAMS INPUT TAPE SEGTRU - BEGINNING REGION AND ZONE NOT FOUND

   Fatal error - start region and zone specified on LEM control card, LEM 02, is not present in file.

2. CAMS INPUT TAPE SEGTRU - ENDING REGION AND ZONE NOT FOUND

   Warning - end region and zone were not found, so CAMS processed all records until EOF (end-of-file).

3. CAMS INPUT TAPE XXXX - MISSING RECORD

   Fatal error - input tape does not correlate correctly with key tape SEGTRU - perhaps wrong file mounted for SEGID or this input tape.

4. BETA DISTRIBUTION ERROR - FLAG = X

   Warning - Beta distribution subroutine, BETAD, returns error for mean production error:

   a. FLAG = 1 — mean not in range $0 \leq$ mean ≤ 1 so if mean > 1, mean set to 1; mean < 0, mean set to 0. (mean = XBAR of Equations B2-2c., C2-2b. in Section 3.3)

   b. FLAG = 2 — sigma not in range $0 \leq$ sigma $\leq \text{XBAR}\sqrt{\dfrac{1-\text{XBAR}}{\text{XBAR}+10^{-4}}}$ so was reset within BETAD.

   c. FLAG = 3 — the random number could not be found within 35 iterations via the inverse incomplete Beta function method, so XI set to XBAR. (See Section 3.3, B2 and C2.)

PART I

PROBLEM DESCRIPTION FOR THE CAS MODULE OF THE LEM PROGRAM

Table of Contents

Problem Description for the CAS Program

1.0 SCOPE

This document describes the requirements and processing logic for the CAS Simulator Module of the LACIE Error Model Program (LEM), which is an integral part of the Large Area Crop Inventory Experiment (LACIE) system.

1.1 PROGRAM CAPABILITIES

The purpose of the CAS Simulator is to model the LACIE aggregation technique including the aggregation of wheat area and production at the stratum, zone, region, and country levels and the estimation of the accuracy of the technique.

The CAS simulator provides the following functions:

(1) Calculate area and production estimates and standard statistics at the stratum, zone, region, and country level.

(2) Calculate the estimated confidence level associated with the 90% accuracy criterion at the country level.

(3) Compute the mean values of the estimates, errors, and variances of area, yield, and production for repetitive Monte Carlo trials.

(4) Calculate the true simulation confidence level associated with the 90% accuracy criterion at the country level based on the statistics of repetitive Monte Carlo trials.

(5) Output the CAS summary reports and CAS Output Files.

The CAS aggregation is performed at various prediction intervals during the growing season. The CAS simulator determines the present interval and obtains the appropriate information from the YES and CAMS output files and the Substrata Historical File.

CAS processes the area data on the substrata level. The substrata Group No. is determined in order to determine the proper aggregation scheme. Group I substrata parameters are computed if any segments have been acquired for the substratum, otherwise it is reassigned to Group III. Group II parameters are computed if any segment has been acquired in the collection of Group II substrata in the stratum, otherwise these substrata are reassigned to Group III. Appropriate historical data is obtained for the Group III substrata.

Once the appropriate parameters have been computed, the Group I, II, and III substrata estimates are made and aggregated to the stratum level. Here they are combined to give the total stratum estimate.

The variances are computed for each group and aggregated to the stratum level and combined. Group I and Group II variances use a least squares fit of the historical vs. estimate data for the stratum as part of the variance computation. In order to compute the within-county variance estimates, all of the substrata (counties) within a zone are grouped into homogeneous classes and the within-class variance estimates are obtained by the least squares fit and then used for the within-county variance estimate.

The strata yield data from the YES Output File is combined with the area data to determine production. Area and production are then aggregated to the zone, region, and country levels. Production variance is also computed and aggregated along with the area variance. The estimated confidence level is computed from this variance data and the mean value at the country level for area and production.

The simulation also keeps track of the error between the LACIE estimate and the true value. These errors are computed for area, yield, and production at various levels of aggregation. During successive Monte Carlo trials, these values and other appropriate parameters are accumulated to enable computation of the simulation mean and variance of each parameter and error at various levels of aggregation. The Monte Carlo statistics are used to compute the simulation confidence level about the true mean.

The results of the simulation are output in the CAS summary output reports. The data is also maintained on the CAMS Cumulative and Distribution Output Files for further processing by the POUT Output Processor if required.

1.2 PROGRAM DEVELOPMENT AND ORGANIZATION

See Section 1.2 of the Problem Description for the LEM program.

1.3 OPERATIONAL ASSUMPTIONS

See Section 1.3 of the Problem Description for the LEM program.

2.0 INPUT

One control card set and three input files are required as inputs to the CAS simulator. The control card set specifies parameters and flags which control the execution of the CAS simulator.

Data files are used to input data to CAS from the YES and CAMS modules within LEM and from the LUMP program.

2.1 CARD INPUT

Three control cards are required by the CAS simulator. The first contains various flags and parameters. The second and third control cards specify the prediction dates for which the CAS computations are to be performed.

2.1.1 List of Data Quantities

See the Input Data Description sheet on Page 5.

2.1.2 Card Formats

Each CAS control card has a fixed field format as shown in Figure 2.1.

"CAS" is entered in Columns 75-77 of each control card, and a sequence number is entered in Columns 79-80.

2.1.3 Deck Setup

Each of the three CAS control cards is required (even if only seven or less prediction points are specified), and they must be in the proper order. Furthermore, the CAS control cards must follow the LEM control cards and the CAMS control cards as specified in Section 2.1.3 of the Problem Description for the LEM program.

2.1.4 Rules for Entering Data on Cards

1. Integers must be right justified.
2. The prediction dates must be entered in the format

   7 (3I2, 1X)

   with a maximum of seven dates per card.

INPUT DATA DESCRIPTION

ard ol.	Name	Dimension	Nominal Value	Range	Description
1-3	NHISTY	1	—	1-20	M = Number of historical years for Group III ratio calculations. (No longer used; Set = 1)
4-6	H	1	—	3-99	H = Minimum number of segments required for applying S^2 regression equation.
7-9	TOPT	1	0	0, 1	T - option flag: = 0 to set T = 0, = 1 to calculate T where T is the second term of the variance equation for $\hat{V}_{2S}$.
10-12	AUNITS	1	0	0, 1	Units Option: = 1 to print area in hectares and production in metric tons, = 0 to print area in acres and production in bushels
13-15	DISTFF	1	0	0, 1	CAS distribution file flag: = 0 to generate CAS distribution file, = 1 otherwise
16-27	IWIND	4	0	0, 1	Prediction bio-window flags: IWIND(n) = 1 to process bio-window n, = 0 otherwise
28-39	WPRIOR	4	0	0-4	Bio-window priorities: List of bio-windows in decreasing order of priority. e.g., 4, 1, 3, 2 or 3, 1, 0, 0
40-42	APREP	1	0	0, 1	Print option for area and production summary report: = 1 to print report, = 0 otherwise
1-48	IPRD	3, 14	0	>64 year 01-12 month 1-31 day	Prediction dates (up to 14 dates): IPRD (1, n) = year - 1900 IPRD (2, n) = month IPRD (3, n) = day The prediction dates must be in ascending order. The first zero date terminates the list.

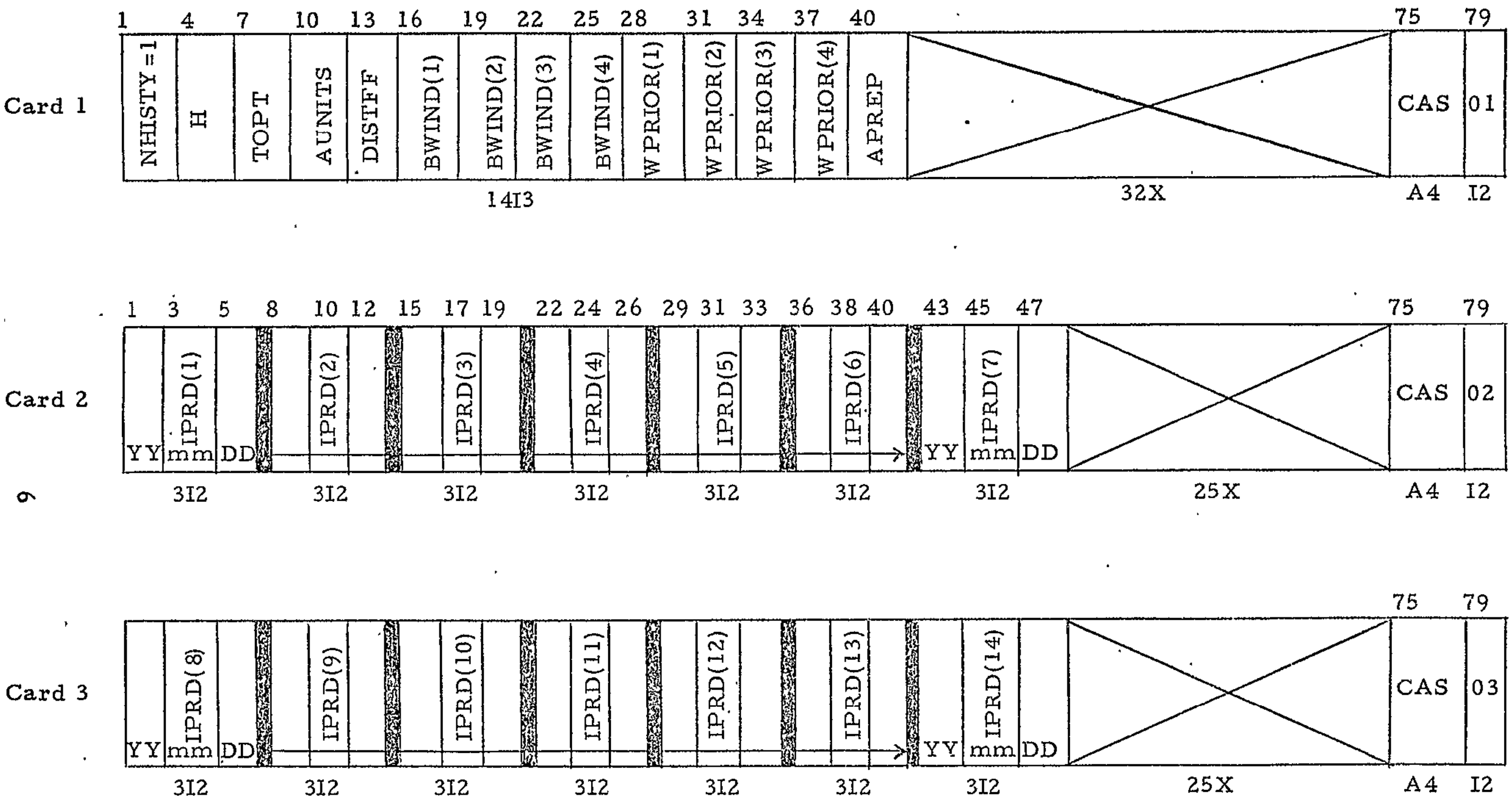


Figure 2-1. Data Card Formats

2.2 INPUT FILES

The following files are required as inputs to CAS.

YES Output File	-	Strata yield data from the YES module
Substrata Historical File	-	Substrata information generated by LUMP
CAMS Output Data	-	Segment data generated by the CAMS module

In addition, on a restart run the CAS Cumulative File and on option, the CAS Distribution run must be input since the data on those files is accumulated over all Monte Carlo iterations.

3.0 PROCESSING

3.1 OVERVIEW

The CAS simulator module is divided into two major subdivisions. The first subdivision consists of two subroutines, CASIN and CASER1, which process the CAS control cards and write input error messages, respectively. The remaining portion of CAS executes as an overlay subprogram under the control of the LEM driver.

3.2 PROGRAM FLOW

Flow diagrams for the CAS simulator are presented in Figures 3-1 through 3-5. Figure 3-1 is an overall flow chart with very little detail. It represents the CAS driver. Figure 3-2 shows the detail of the substrata classification logic. Figure 3-3 shows the detail of the first pass CAS logic, which generates data sets 1-9 for a given bio-window or prediction date. Figure 3-4 shows the detail of the second pass logic, which computes the area variances for all strata with acquired segments. Figure 3-5 shows the detail of the third pass CAS logic, which generates data sets 10-17, and 19 for the same bio-window or prediction date.

3.3 PROCEDURES AND EQUATIONS

The symbols used in this section are defined in Appendix A. The data set descriptions are given in Appendix B. The equations are given in Appendix C.

As seen in the flow diagrams, the CAS logic consists of the following tasks:

- Initialization
- Determination of substrata classes
- First pass calculations
- Second pass calculations
- Third pass calculations
- Report generation

3.3.1 Initialization

The general initialization tasks are performed by subroutine CASINT.

1. Rewinding all input files

2. Initializing flags and counters

3. Determining number of records to skip on files YESOUT, CAMSF, and SUBHST in order to position them at the proper starting region and zone (STARTR and STARTZ).

3.3.2 Determination of Substrata Classes

The determination of the substrata classes is performed in subroutine CLASSN, which is called by subroutine CASPP on the first Monte Carlo iteration of each computer run for each bio-window or prediction date.

1. Substrata historical data is read from either the SUBHST file or the ISUBH2 file. If the first bio-window or prediction point is being processed, then the substrata historical file SUBHST is used. Otherwise, the intermediate substrata historical file ISUBH2 is used.
2. Each zone in the country is processed one at a time. Within each zone the collection of substrata is partitioned into one or more homogeneous classes of substrata. By this partitioning process a class number is assigned to each substrata in the zone.
3. After each zone is partitioned and a class number is assigned to each substrata within that zone, the substrata data along with the assigned class number is written back onto the ISUBH2 file.
4. The details of the partitioning process are given in the writeup of subroutine CLASSN.

3.3.3 First Pass Calculations

The first pass calculations are performed in subroutine CASPP, which is called for each bio-window or prediction date.

1. Correctly position the files YESOUT, CAMSF, and SUBHST at the proper starting region and zone.
2. Read the strata yield data from the YESOUT file.
3. Read the substrata historical data from the SUBHST file.
4. The substrata group number is examined and if it is 1 or 2, the segment data is read from the CAMSF file.

5. Depending upon the substrata group number, the segment data is aggregated into data set 1 or 2, or data set 3 is generated.

6. Again depending upon the substrata group number, the substrata data (data set 1, 2, or 3) is aggregated into data set 4, 5, or 6.

7. If no Group II segments are acquired in the strata, then data set 5 is added to data set 6 and data set 5 is cleared to zero.

8. The strata data in data sets 4, 5, and 6 is aggregated into data set 7 at the zone level.

9. The zone data in data set 7 is then aggregated into data set 8 at the region level.

10. The region data in data set 8 is aggregated into data set 9 at the country level.

3.3.4 Second Pass Calculations

The second pass calculations are performed in subroutine CAS2, which is called by CASPP.

1. If no segments were obtained for the entire country, the message NO ACQUISITIONS IN COUNTRY is printed out and the rest of the logic is skipped.

2. The region, zone, strata, and substrata pointers for the direct access files CASDSF and ISUBH2 are initialized.

3. The next record (data set 8) from file CASDSF is read into memory.

4. The next zone record (data set 7) from file CASDSF is read into memory.

5. The next strata record (data sets 4, 5, 6) from file CASDSF is read into memory.

6. The next substrata record from file ISUBH2 is read into memory.

7. If the substrata class number is zero, if there are less than two acquired segments in the zone, or if there are no acquired segments in the strata, then the rest of the substrata computations are skipped (steps 8-10).

8. Next the group number is tested. If it is 3, then the rest of the substrata computations are skipped.

9. If the group number is 2, then M_{2j}, the number of acquired Group II segments in the strata is tested. If it is zero, then the rest of the substrata computations are skipped. If $M_{2j} > 0$, then the computation of the substrata variance multiplier is completed by multiplying VMULTK by WA_{2S}/M_{2j}.

10. If the group number is 1 or 2, then VMULTK is multiplied by S^2 for the proper substrata class and the result is added to the quantity V1V2S = $V_{1S} + V_{2S}$.

11. Steps 6-10 are performed for each substrata in the stratum.

12. After all of the substrata in the strata are processed, M1K2KZ, the number of acquired segments in the zone is examined. If it is less than 2, then steps 13-17 are skipped. If M1K2KZ ≥ 2, then the number of acquired segments in the stratum is examined. If there are no acquired segments in the stratum, then steps 13-17 are skipped.

13. The term T is added to V1V2S = $V_{1S} + V_{2S}$ and to the group 2 analytic area variance ANVS2.

14. The quantity

$$\tau_S^{\,2} = \left[1 + \frac{\widetilde{WA}_{3S}}{\widetilde{WA}_{1S} + \widetilde{WA}_{2S}}\right]^2$$

is computed.

15. Finally, the area variance $V_S = \tau_S^{\,2}\,(V_{1S} + V_{2S})$ and the analytic area variance ANVARS = $\tau_S^{\,2}$ (ANVS1 + ANVS2) are computed for the current stratum, which has at least one acquired segment.

16. The strata record (data sets 4, 5, 6) is then written back onto the CASDSF file.

17. Also, the terms $V_{1S} + V_{2S}$ are added to the sum

$$ESTVZ = \sum_{zone} (V_{1S} + V_{2S})$$

and the terms ANVS1 + ANVS2 are added to the sum

$$ANALVZ = \sum_{zone} (ANVS1 + ANVS2)$$

18. If there were no acquired segments in the stratum or if there were less than two acquired segments in the zone, then the historical wheat area for the stratum is added to HWAZ3.

19. Steps 5-18 are performed for each stratum in the zone. When the last stratum of the zone has been processed, the zone data record (data set 7) is written back onto file CASDSF.

20. Then if there were at least two acquired segments in the zone, ESTVZ is added to ESTVR and ANALVZ is added to ANALVR.

21. Steps 4-20 are performed for each zone in the region.

22. If there were any zones in the region with at least two acquired segments, then the region data record (data set 8) is written back onto the CASDSF file.

23. Also, ESTVR is added to ESTVC and ANALVR is added to ANALVC.

24. Steps 3-23 are performed for each region in the country.

25. When all regions have been processed, control is returned back to subroutine CASPP, which called CAS2.

3.3.5 Third Pass Calculations

The third pass calculations are performed in subroutine CAS3, which is called by CASPP.

1. If no segments were acquired for the entire country, then a return to CASPP is performed.

2. Data from the zone, region, and country levels (data sets 7, 8, and 9) is combined with the strata data in data sets 4, 5 and 6 to generate data set 10 at the strata level.

3. The strata data in data set 10 is aggregated up to the zone level (data set 11). It is also accumulated in data set 14 of the CAS Cum file over all Monte Carlo iterations.

4. If the print flag is set (as directed by the LEM input parameter (IPRCAS), the strata portion of the Area and Production Summary report is printed out.

5. Next the zone data in data set 11 is aggregated up to the region level (data set 12). It is also accumulated in data set 15 of the CAS Cum file and is entered into the CAS Distribution file (data set 19).

6. If the print flag is set, the zone portion of the Area and Production Summary report is printed out.

7. Then the region data in data set 12 is aggregated up to the country level (data set 13). It is also accumulated in data set 16 of the CAS Cum file and is entered into the CAS Distribution file (data set 19).

8. If the print flag is set, the region portion of the Area and Production Summary report is printed out.

9. Finally the country data in data set 13 is accumulated in data set 17 of the CAS Cum file and is entered into the CAS Distribution file (data set 19).

10. If the print flag is set, then the country data for this bio-window or prediction date is saved for the Country Summary report to be printed after all bio-windows and prediction dates have been processed.

11. Also if the print flag is set, the country portion of the Area and Production Summary report is printed out.

3.3.6 Report Generation

The Area and Production Summary report is printed out during the second pass as each strata, zone, region, and country is processed.

The Country Summary report is printed out after all bio-windows and prediction dates have been processed.

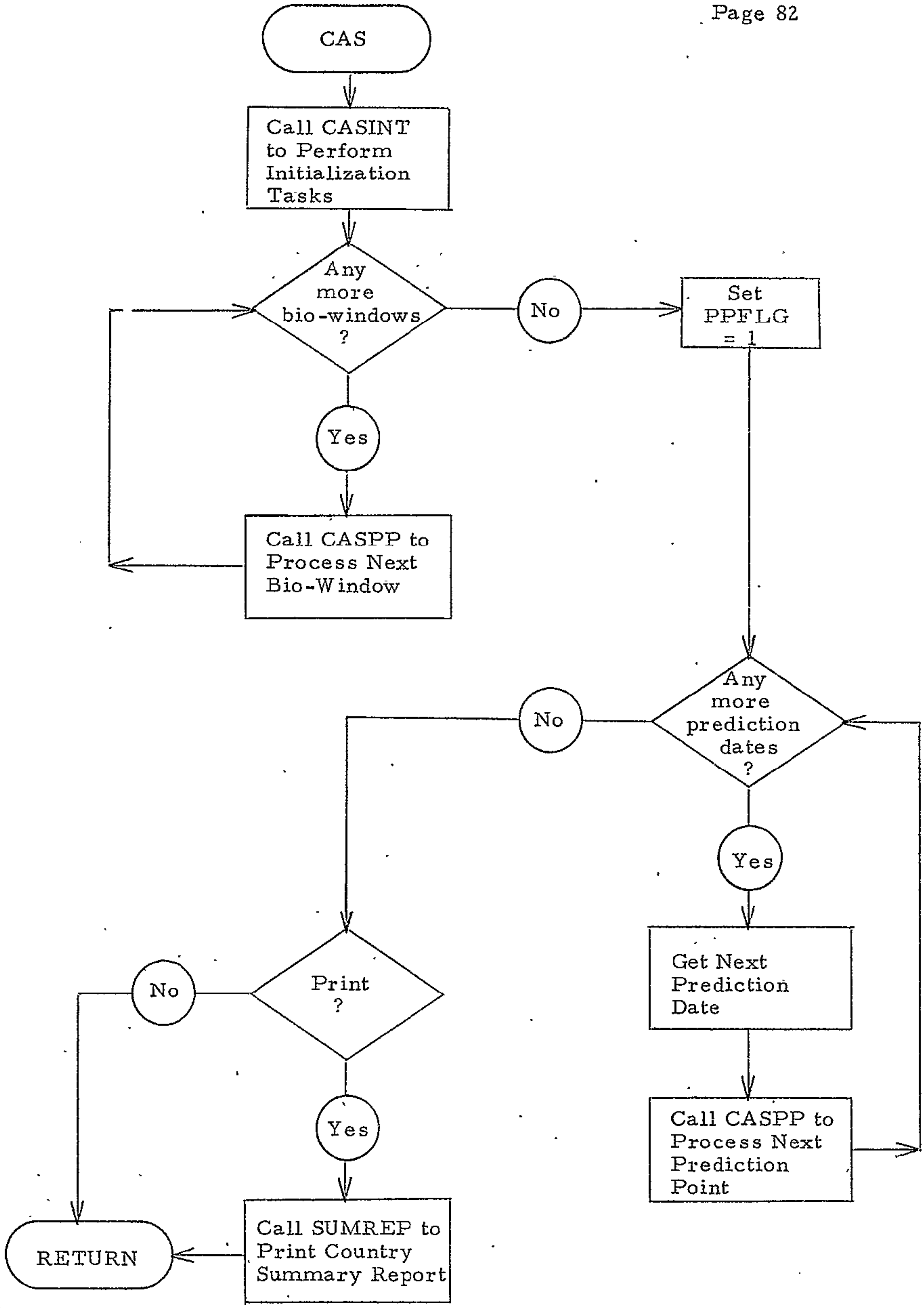


Figure 3-1. CAS Flow Diagram

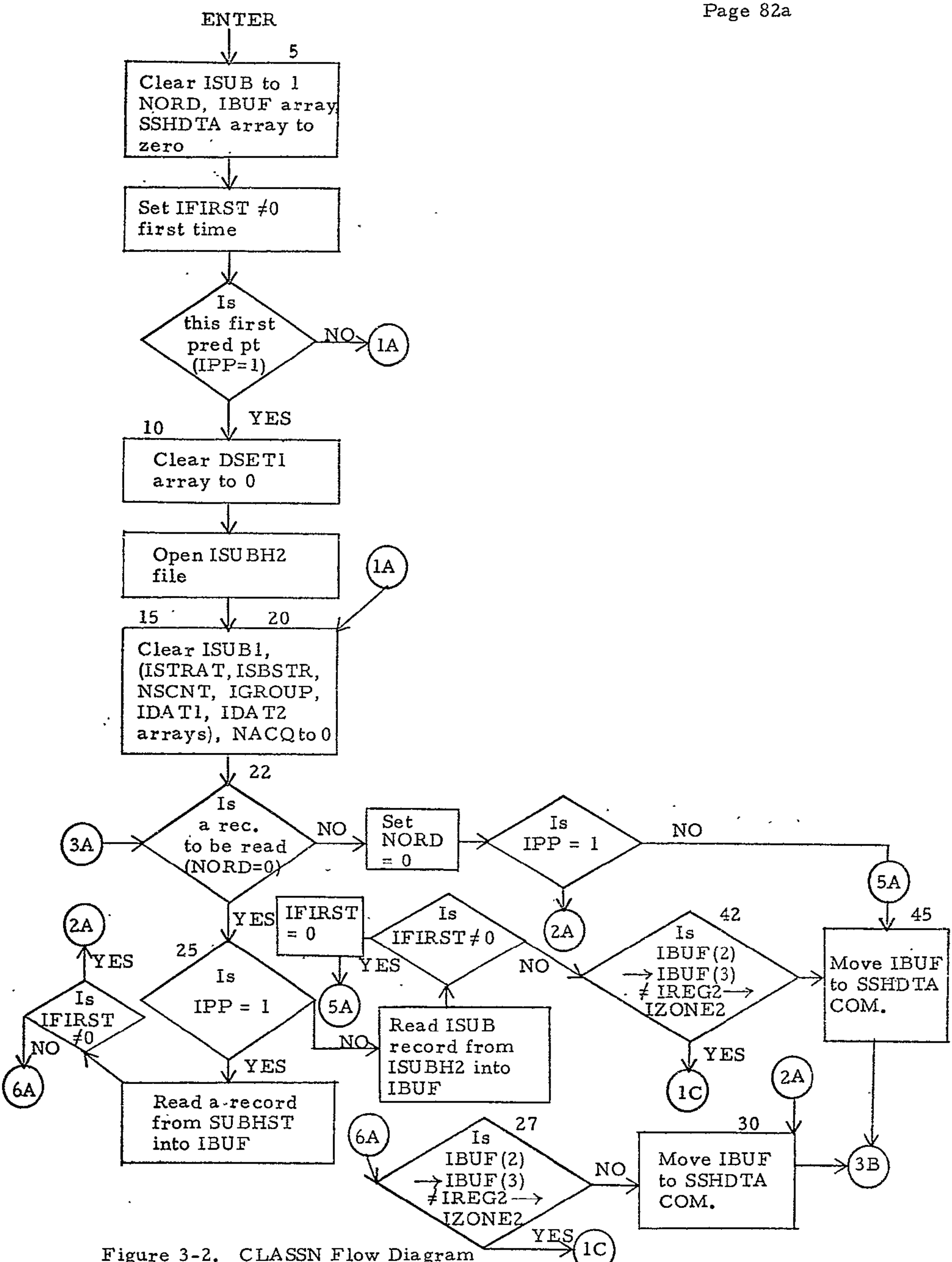


Figure 3-2. CLASSN Flow Diagram
(Sheet 1 of 4)

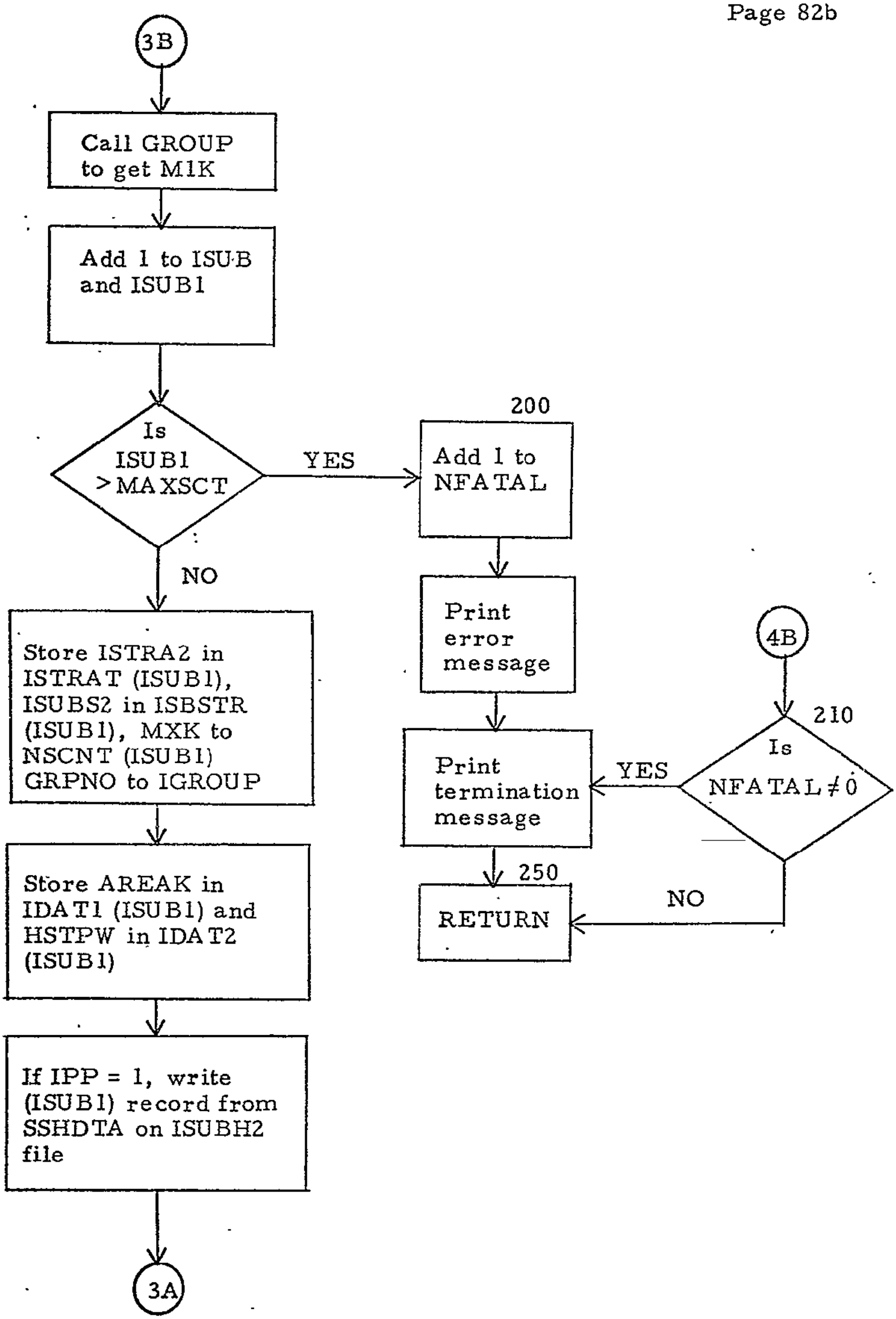


Figure 3-2. CLASSN Flow Diagram (Sheet 2 of 4)

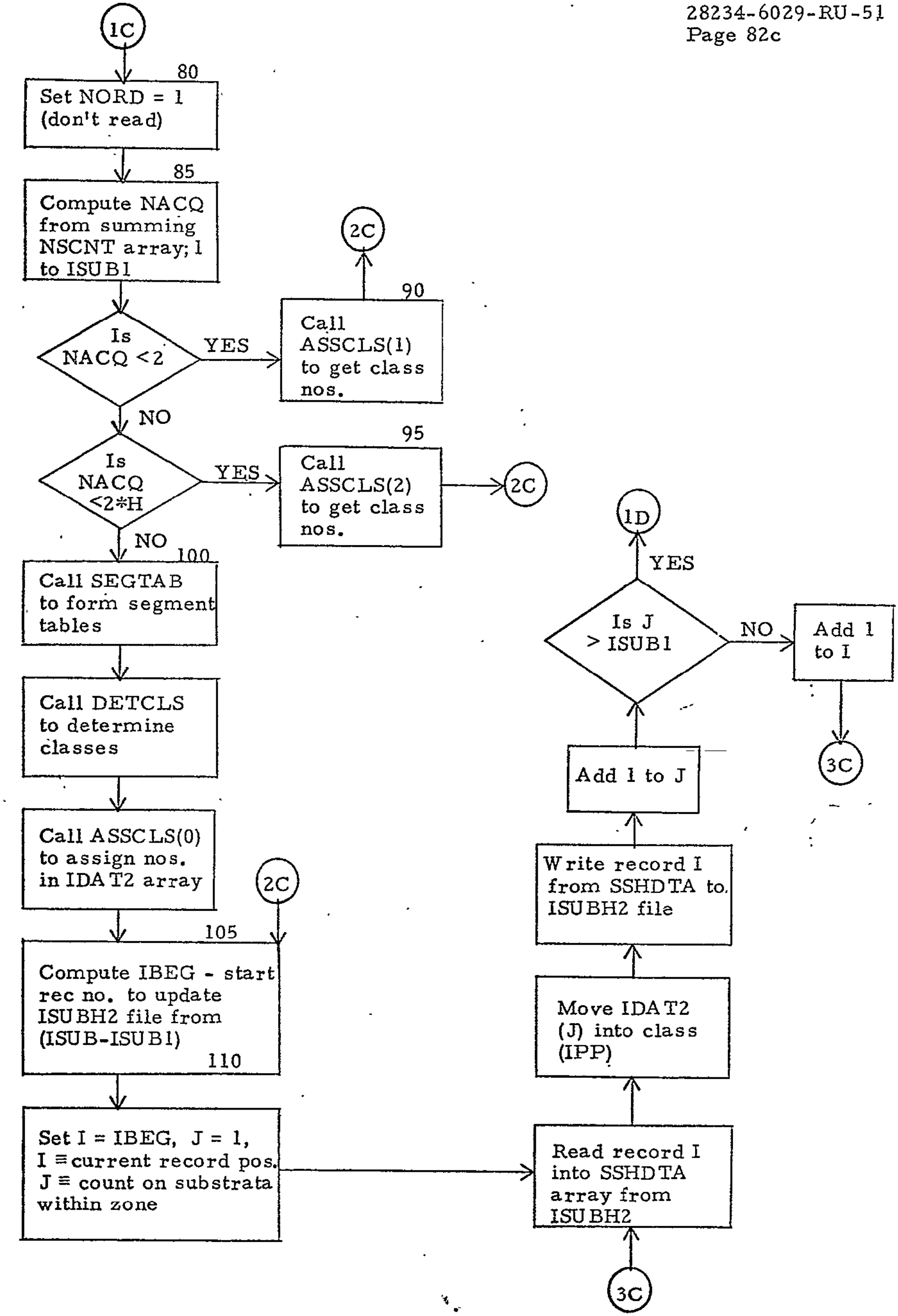


Figure 3-2. CLASSN Flow Diagram (Sheet 3 of 4)

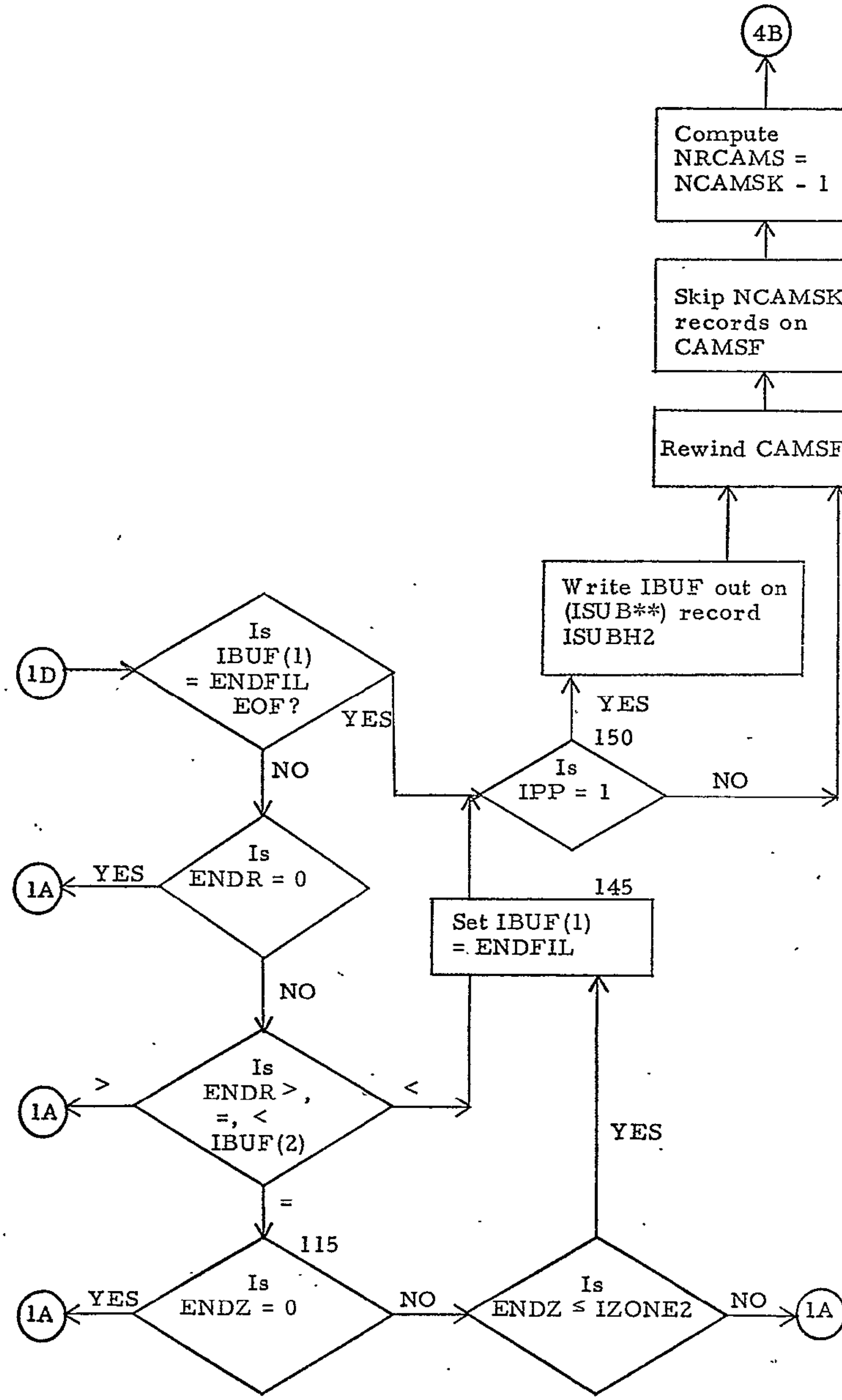


Figure 3-2. CLASSN Flow Diagram (Sheet 4 of 4)

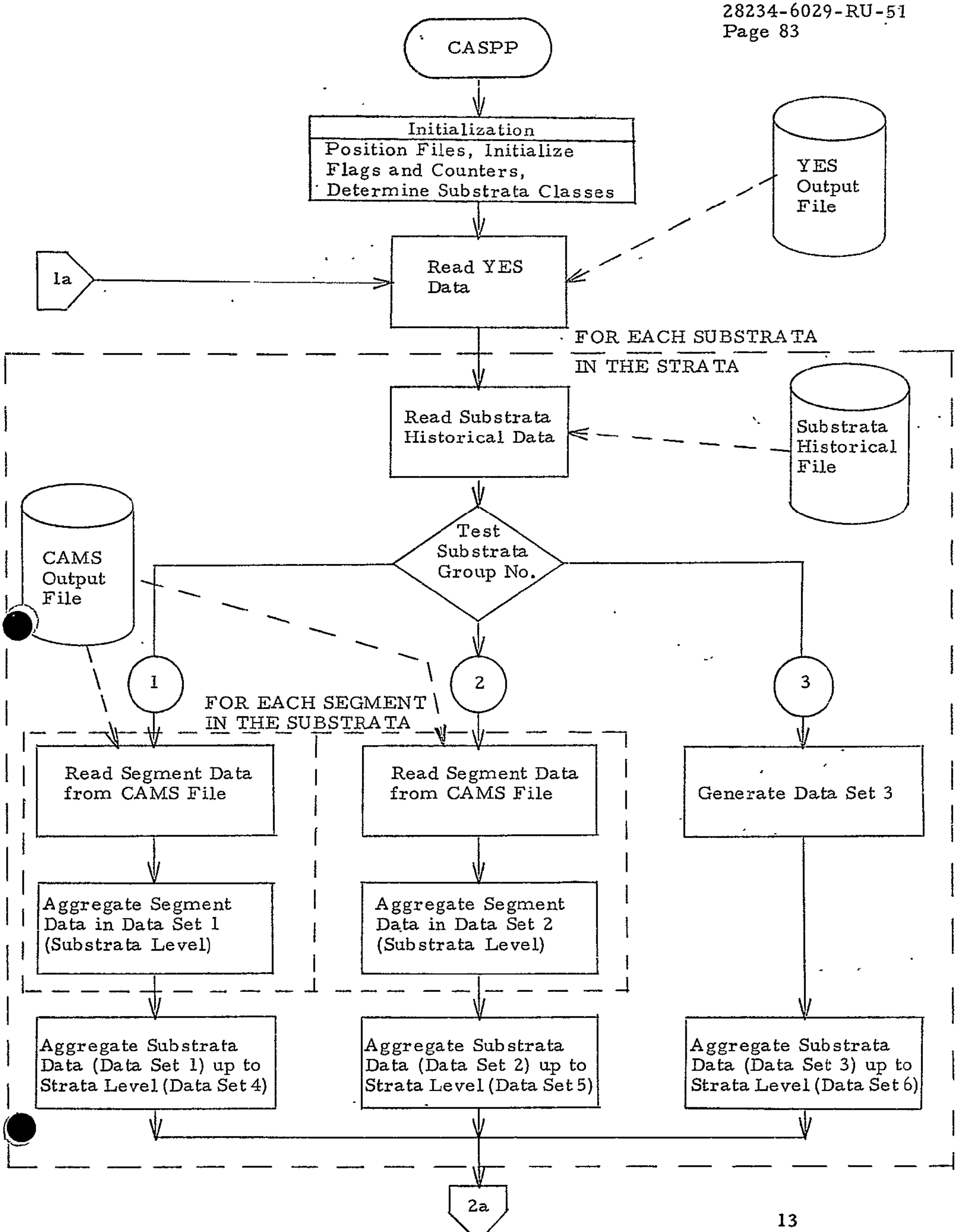


Figure 3-3. CASPP Flow Diagram (Sheet 1 of 2)

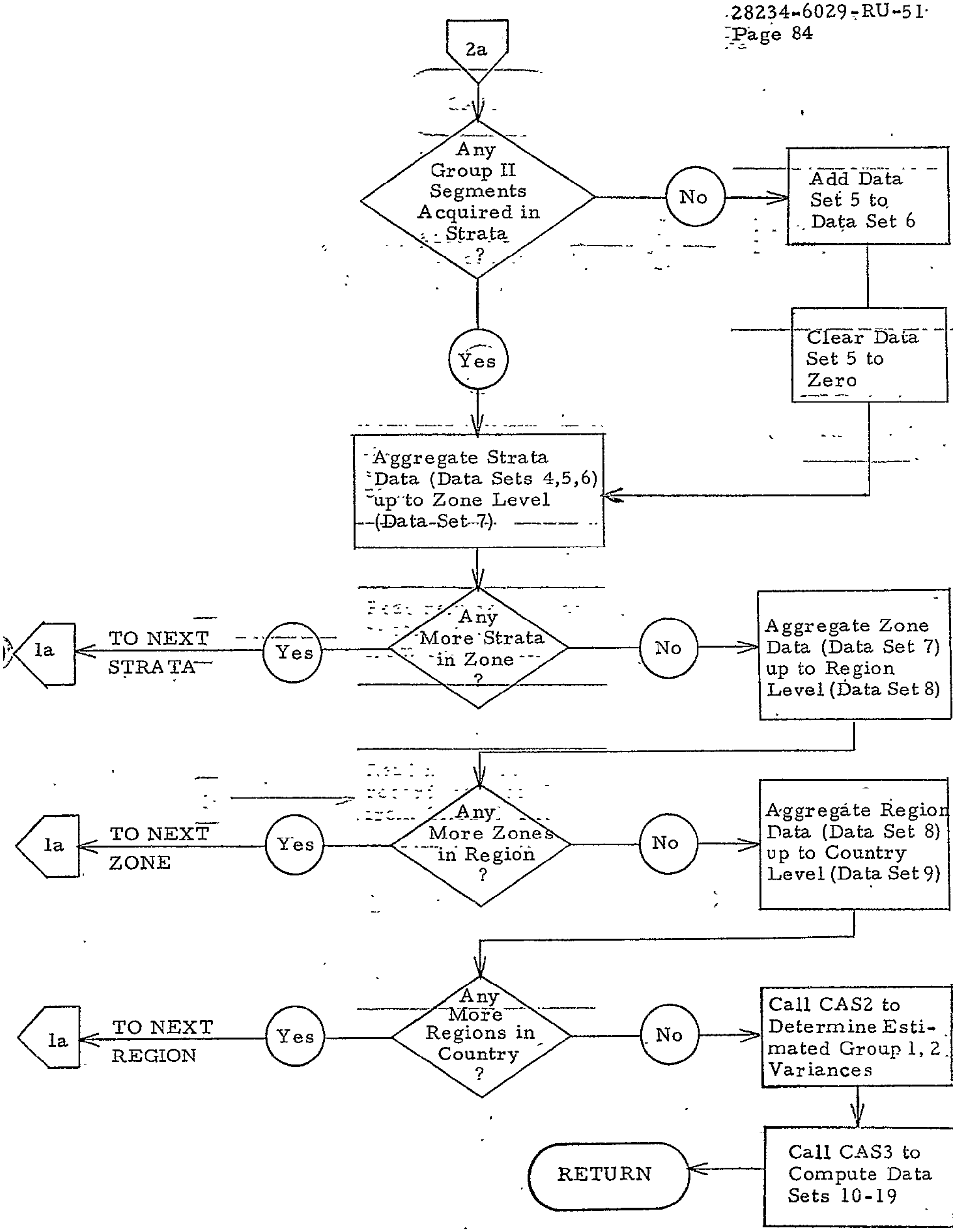


Figure 3-3. CASPP Flow Diagram (Sheet 2 of 2)

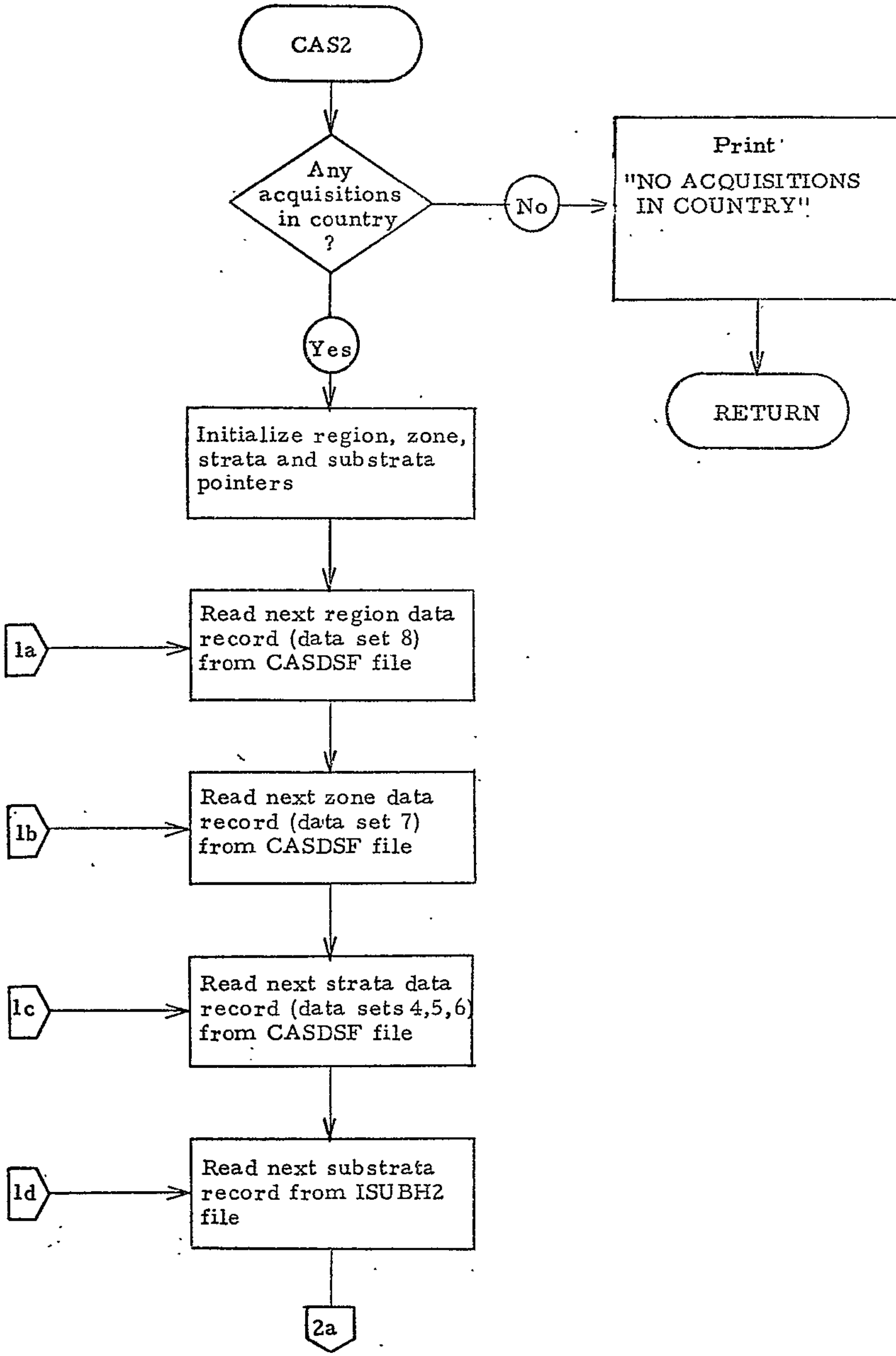


Figure 3-4. CAS2 Flow Diagram (Sheet 1 of 5)

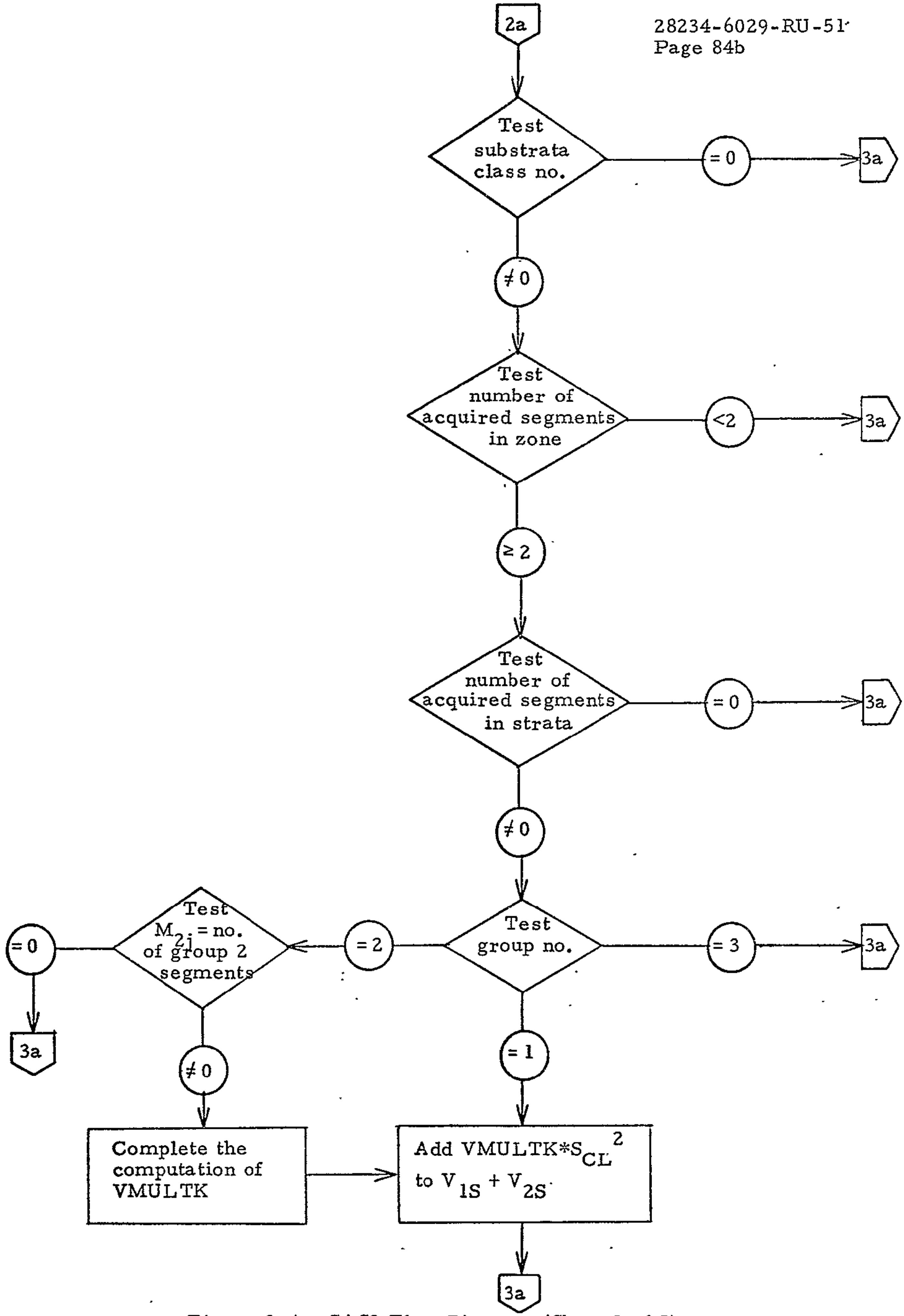


Figure 3-4. CAS2 Flow Diagram (Sheet 2 of 5)

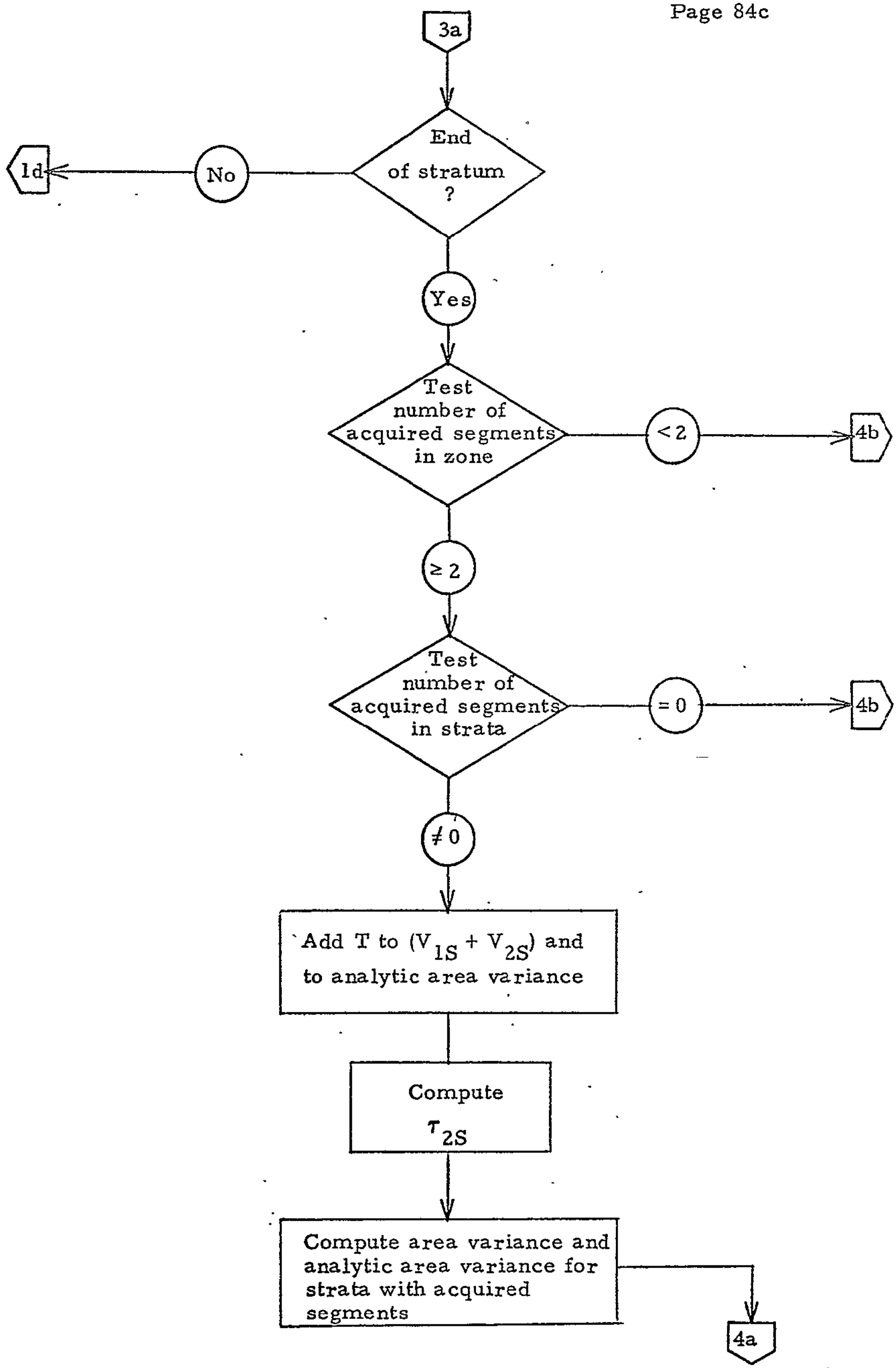


Figure 3-4. CAS2 Flow Diagram (Sheet 3 of 5)

C-2

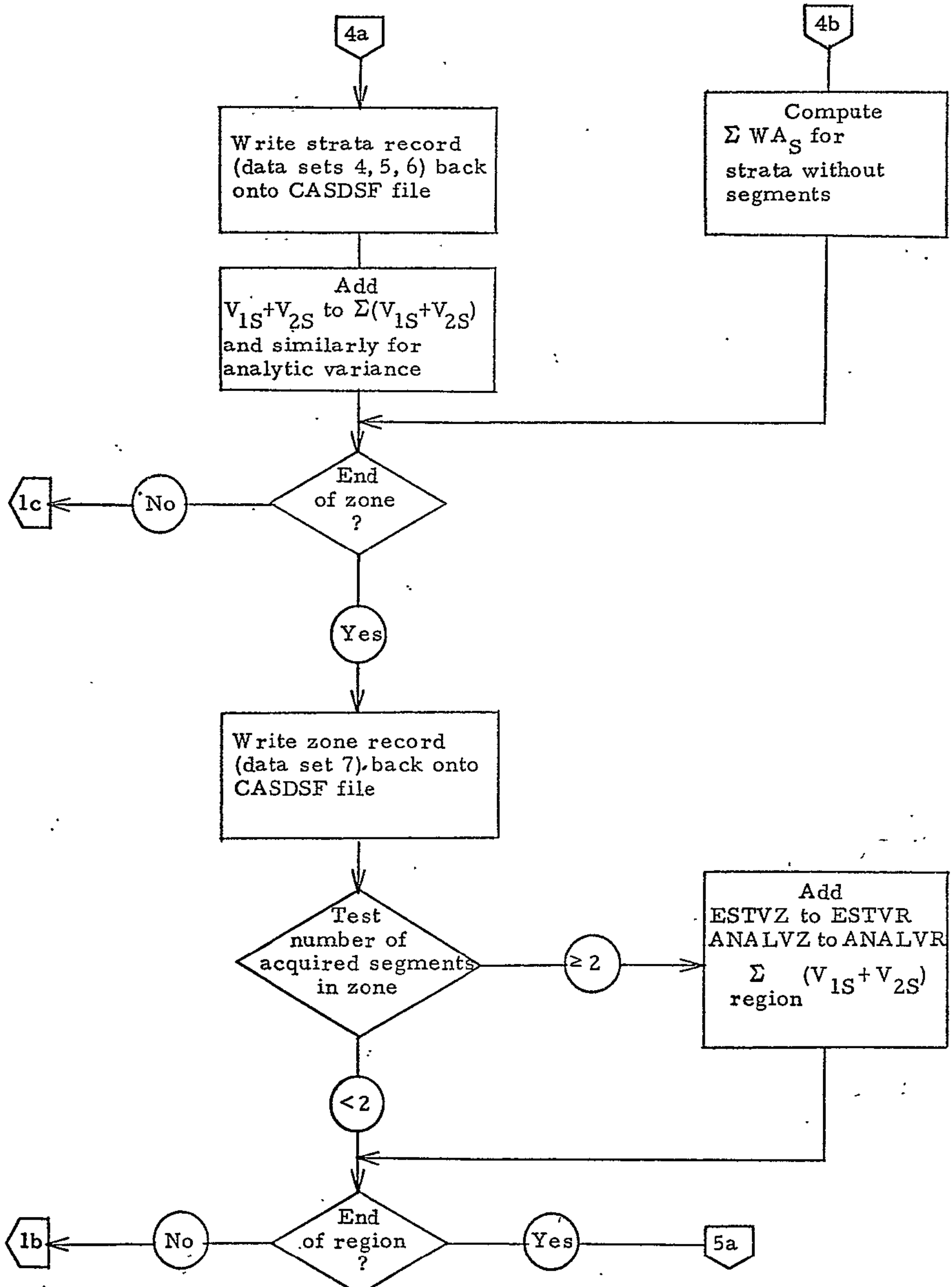


Figure 3-4. CAS2 Flow Diagram (Sheet 4 of 5)

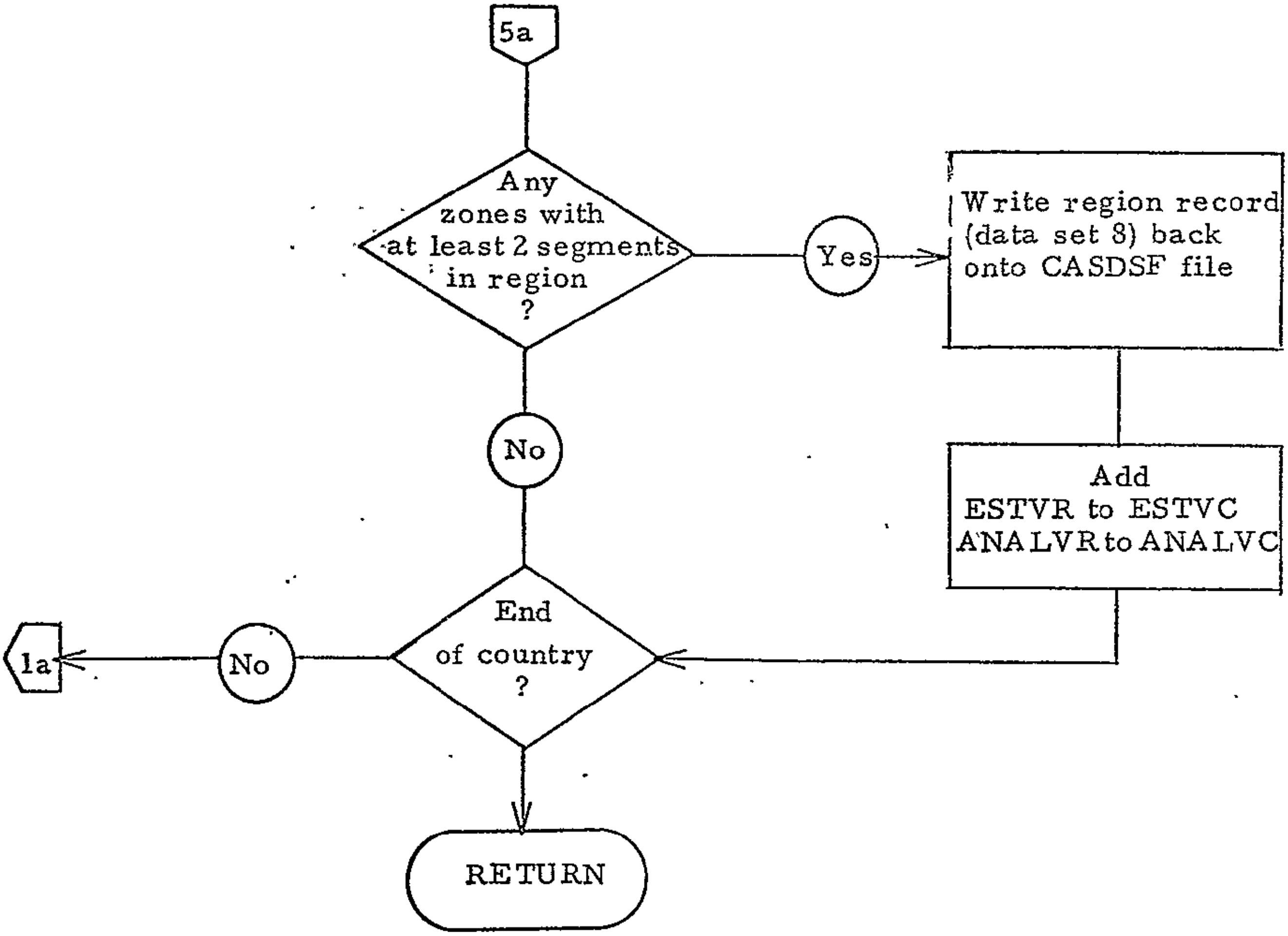


Figure 3-4. CAS2 Flow Diagram (Sheet 5 of 5)

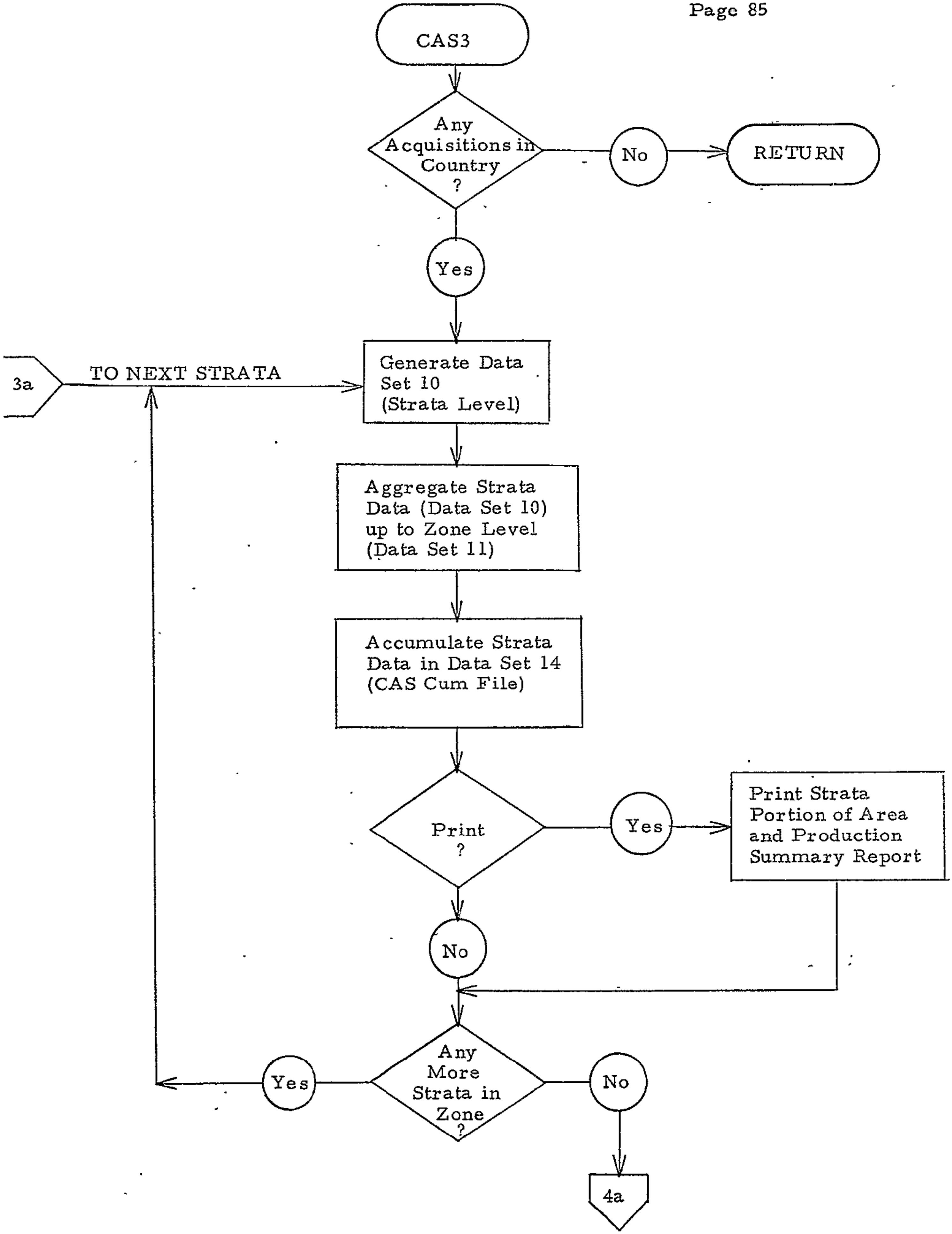


Figure 3-5. CAS3 Flow Diagram (Sheet 1 of 4)

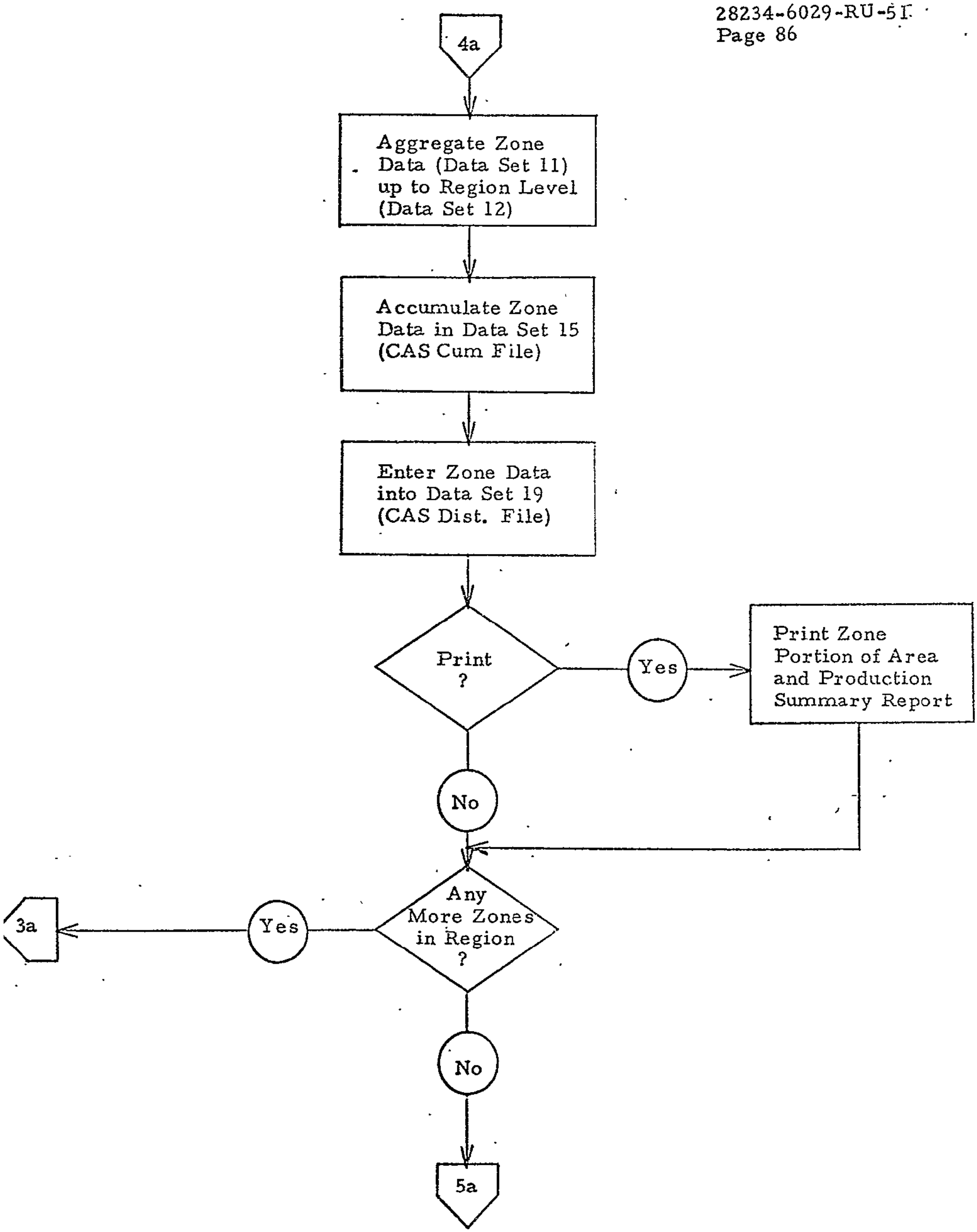


Figure 3-5. CAS3 Flow Diagram (Sheet 2 of 4)

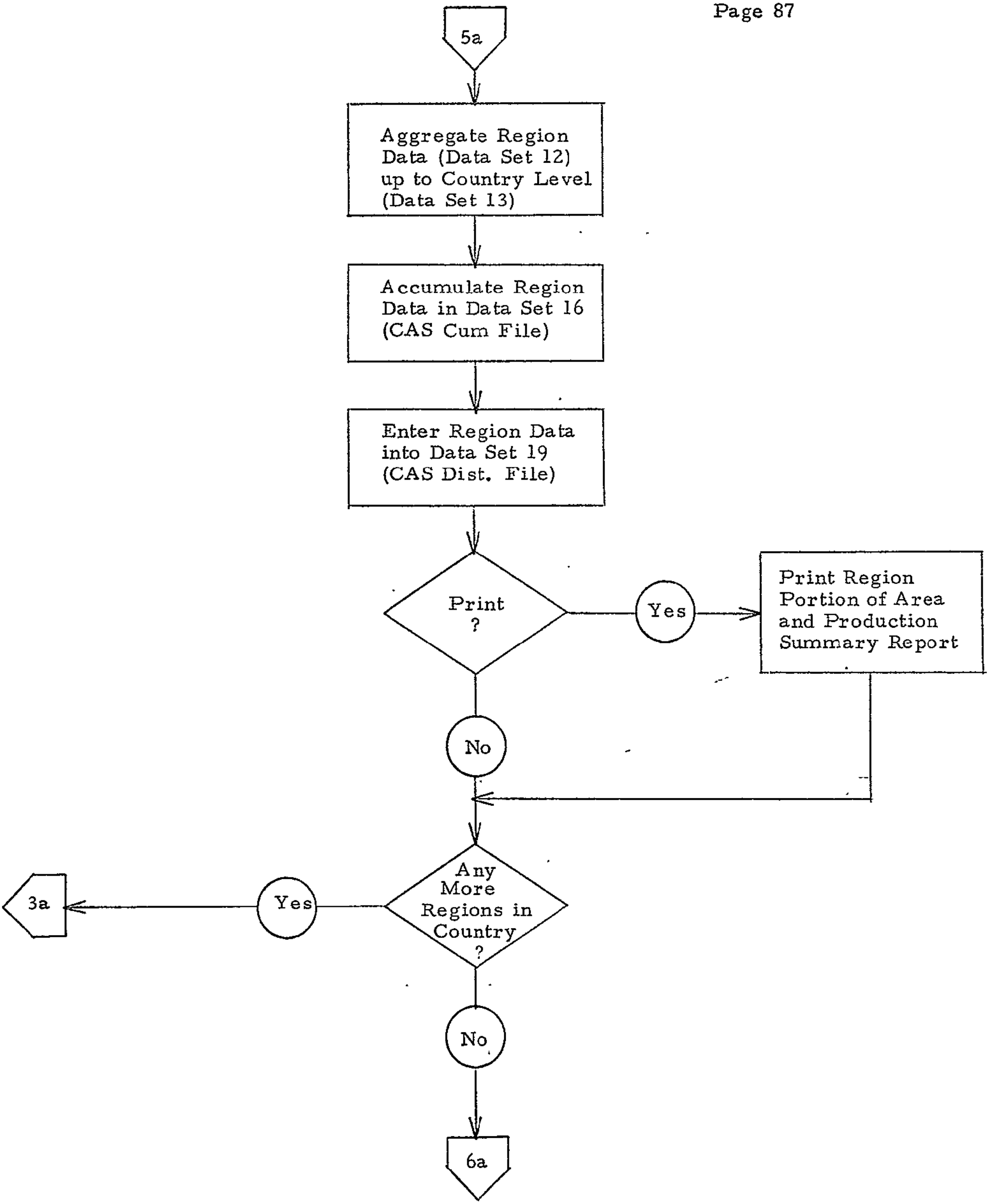


Figure 3-5. CAS3 Flow Diagram (Sheet 3 of 4)

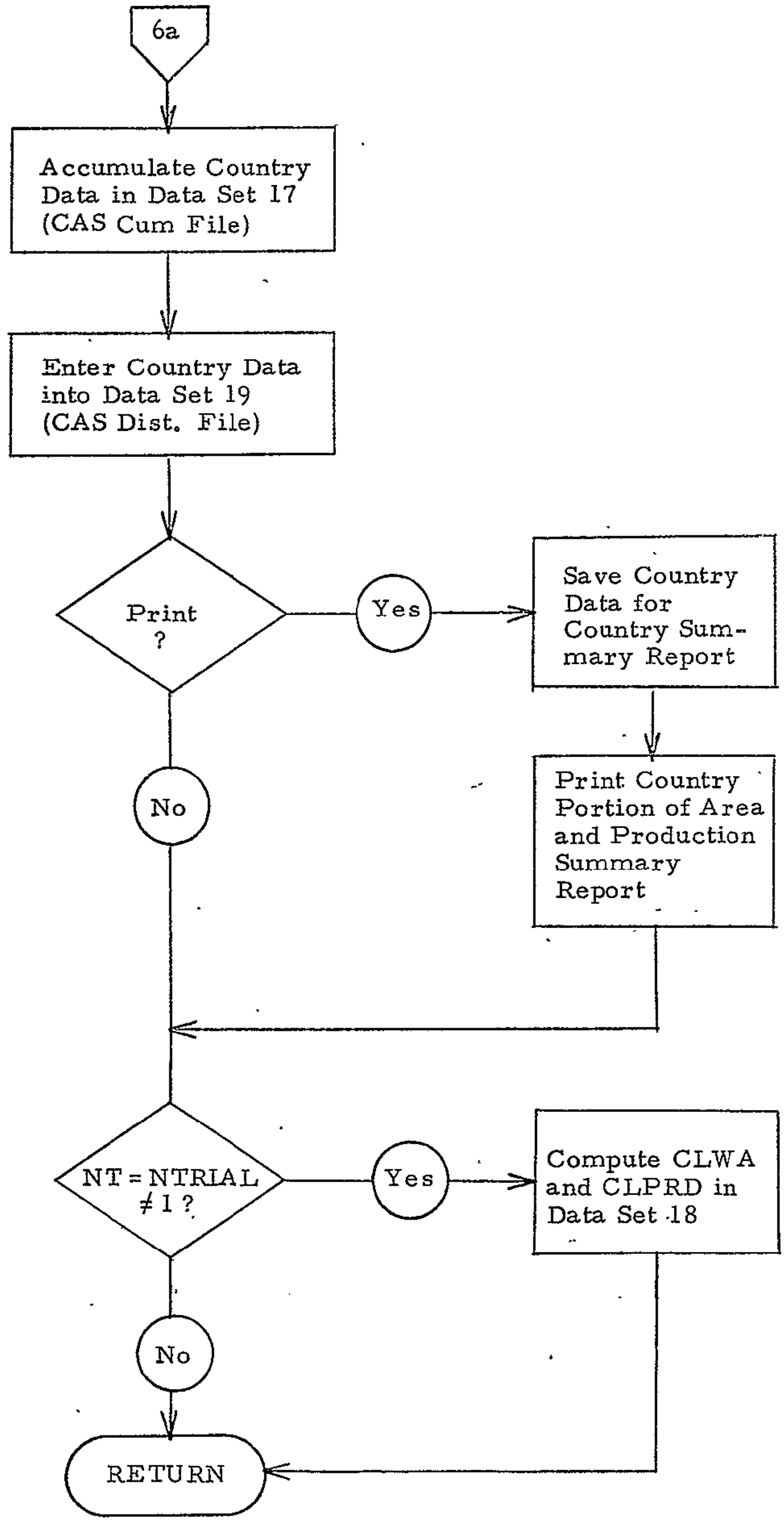


Figure 3-5. CAS3 Flow Diagram (Sheet 4 of 4)

4.0 OUTPUT

4.1 PRINTED DATA

4.1.1 Printed Reports

Two reports, the Area and Production Summary Report and the Country Summary Report, may be generated by the CAS Simulator under the control of the IPRCAS parameter on the LEM control cards and the APREP parameter on the CAS control cards. If APREP = 1, then the Area and Production Summary Report will be produced on each iteration that the Country Summary Report is produced. If APREP = 0, then the Area and Production Report will not be generated at all. The Country Summary Report may be produced on each Monte Carlo iteration, on the first and last iterations only, on the last iteration only, or not at all, depending upon the setting of the IPRCAS parameter (see Section 2.1 of the Problem Description for the LEM Program).

The format of the Area and Production Summary Report is shown in Figure 4-1, and the format of the Country Summary Report is shown in Figures 4-2 and 4-3.

4.1.2 Intermediate Debug

At the present there is no Intermediate Debugging printout specified. However, during checkout the contents of the various data sets will be printed out as they are generated.

4.1.3 Status Information

During the execution of the CAS Simulator, miscellaneous status information will be collected and passed on to LEM for printing out at the end of the run. In particular, the number of records read from the Input Files and the number of records written onto the CAS Cumulative Output File and the CAS Distribution Output File will be saved for printing.

4.1.4 Echo Print Input Card Images

The data specified on the CAS control cards is always printed out in a format that is almost identical to the format on the input card images. Due to differences in the FORTRAN read and write formats,

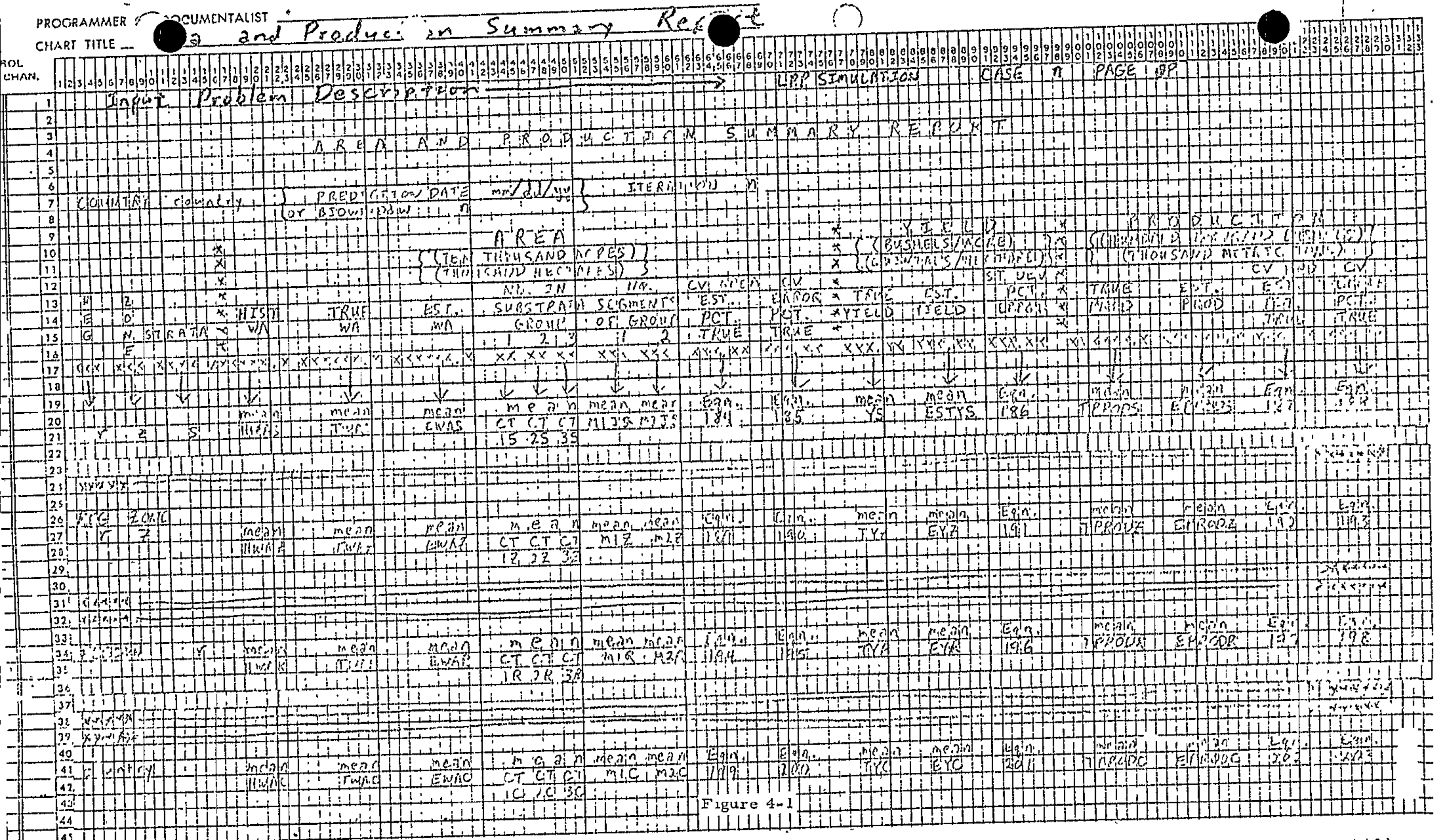


Figure 4-1

ORIGINAL PAGE IS OF POOR QUALITY

PROGRAMMER OR DOCUMENTALIST

CHART TITLE Country Summary Report (Page 1)

LPT SIMULATION CASE n PAGE p

Input Problem Description

C O U N T R Y   S U M M A R Y   R E P O R T

COUNTRY country ITERATION n

		A R E A			Y I E L D		P R O D U C T I O N		
		(TEN THOUSAND ACRES) or (THOUSAND HECTARES)			(BUSHELS/ACRE) or (QUINTALS/HECTARE)		(HUNDRED THOUSAND BUSHELS) or (THOUSAND METRIC TONS)			
PREDICTION POINT	EST. WA	CV ANAL WA (PCT TRUE)	CV AREA EST (PCT TRUE)	CV ERROR (PCT TRUE)	EST. YIELD	ST. DEV PCT ERROR	EST. PRD.	CV ANAL PRDT (PCT TRUE)	CV PRD EST (PCT TRUE)	CV ERROR (PCT TRUE)
		XXX.XX	XXX.XX	XXX.XX	XXX.XX	XXX.XX	XXXXXX.X	XXX.XX	XXX.XX	XXX.XX
1	XXXXXX.X									
2										
3										
4										
mo/da/yr	mean EWAC	Eqn. 204	Eqn. 197	Eqn. 200	mean EYC	Eqn. 201	mean EPPDC	Eqn. 205	Eqn. 202	Eqn. 203
12 10 11 Prediction dates										
mo/da/yr										

HISTORICAL AREA XXXXXX.XX TEN THOUSAND ACRES or THOUSAND HECTARES

TRUE AREA XXXXXX.XX TEN THOUSAND ACRES or THOUSAND HECTARES

TRUE YIELD XXX.XX BUSHELS/ACRE or QUINTELS/HECTARE

TRUE PRODUCTION XXXXXX.XX HUNDRED THOUSAND BUSHELS or THOUSAND METRIC TONS

Figure 4-2

CHART TITLE Country Summary Report (Page 2)

Input Problem Description → LPP SIMULATION CASE n PAGE p

COUNTRY SUMMARY REPORT

COUNTRY country ITERATION n

	AREA CONFIDENCE LEVELS				PRODUCTION CONFIDENCE LEVELS			
PREDICTION POINT	TRUE/ERROR	EST/EST	TRUE/EST	TRUE/WC	TRUE/ERROR	EST/EST	TRUE/EST	TRUE/WC
1	XXX.XXX	XXX.XXX	XXX.XXX	XXX.XXX	XXX.XXX	XXX.XXX	XXX.XXX	XXX.XXX
2								
3								
4								
mo/day/yr	CLWA	mean CLEWA	mean CLATEC	mean CLATWC	CLPRD	mean CLEPRD	mean CLPDTC	mean CLAIWC
	X				X			

X True Confidence Using M/C Error Variance Will Be Output Only on the Final Iteration

Figure 4-3

the printout may be slightly different from the input card images. For example, a blank field will be printed out as -0 rather than being left blank.

4.2 FILES

There are two files output by the CAS Simulator -- the CAS Cumulative Output File, which consists of Data Sets 14, 15, 16, and 17, and the CAS Distribution Output File, which consists of Data Set 19. Both of these files are random access files. The formats and contents of these two files are given in the LACIE File Definition Supplement.

5.0 ERROR PROCESSING

5.1 GENERAL

The program will attempt to find as many errors as possible during the processing of the input control cards. The program will continue checking for additional input errors if any input error is detected. There are two levels of error. These are:

Level 1 - non-fatal, continue processing.

Level 2 - job fatal. Terminate job after processing all input control cards.

When a level 1 error is detected, the program will print an informative message and continue processing. When a level 2 error is detected, the program will print an informative message, set a fatal error flag, and continue processing. When all control cards have been processed, the program will continue executing if no fatal errors were found or will return control back to the operating system if at least one fatal error is detected.

The errors which may be detected by the CAS Simulator are described below.

5.2 INPUT ERRORS DETECTED BY CAS

1. Message:

   IMPROPER LABEL AND SEQUENCE NUMBER ON A CAS CONTROL CARD. LABEL AND SEQ. NO. = __________

   Meaning:

   Fatal error -- the three CAS control cards are supposed to have CAS 0i entered in Columns 75-80 (where i = 1, 2, or 3). Possibly the control cards are out of order. The CAS control cards must always be preceded by the LEM control cards and the CAMS control cards.

Remedy:

Be sure that the LEM, CAMS, and CAS control cards are all present, and are in the proper order. Check the label and sequence numbers to be sure they are entered properly.

2. Message:

NHIST = m IS OUT OF RANGE.
(1 .LE. NHIST .LE. 20)

Meaning:

Fatal error -- NHIST must satisfy $1 \leq \text{NHIST} \leq 20$.

Remedy:

Change the input value of NHIST or change the limits of NHIST within the CAS Simulator (subroutine CASIN). (No longer relevant.)

3. Message:

HH = h IS OUT OF RANGE
(3 .LE. HH .LE. 99)

Meaning:

Non-fatal error -- HH must be within range

$$3 \leq HH \leq 99$$

in order to apply the regression relation for S^2.

Remedy:

Change the input value of HH or change the limits for HH within the CAS Simulator (subroutine CASIN). The program will set HH = 99999 so that the first formula for S_o^2 will always be used.

4. Message:

ILLEGAL WINDOW SPECIFIED IN WPRIOR = W_1, W_2, W_3, W_4
(EACH WINDOW MUST BE 1-4 OR 0)

Meaning:

Fatal error -- an improper value has been specified for one or more of the windows in the array WPRIOR. The only allowable values are 0, 1, 2, 3, or 4.

Remedy:

Correct the offending values.

5. Message:

ALL ENTRIES IN WPRIOR ARE ZERO

Meaning:

Fatal error -- each entry in the array WPRIOR is zero, but at least one window 1-4 must be specified.

Remedy:

Specify at least one non-zero window number in the array WPRIOR.

6. Message:

ILLEGAL PREDICTION DATE yy/mm/dd SPECIFIED. YEAR = yy MUST BE .GE. 64, MONTH = mm MUST BE 1-12, DAY MUST BE 1-31

Meaning:

Fatal error -- an illegal prediction date has been specified in the array IPRD. The prediction date must satisfy

$$\text{year} \geq 64$$
$$1 \leq \text{month} \leq 12$$
$$1 \leq \text{day} \leq 31$$

Note: Dates such as Feb. 30 or Sept. 31 will be accepted by the program without being recognized as being in error.

Remedy:

Correct the offending dates.

7. Message:

PREDICTION DATES NOT IN ASCENDING ORDER OR DUPLICATES.

Meaning:

Fatal error -- the prediction dates entered in the array IPRD must be in ascending order with no duplicates.

Remedy:

Enter the prediction dates in ascending order and eliminate any duplicates.

5.3 PROCESSING ERRORS DETECTED BY CAS

1. Message:

DIVISION BY ZERO NOT ALLOWED
EQN. (n), symbol = 0.

Meaning:

Fatal error -- the program detected a zero divisor in attempting to compute equation (n). The offending zero divisor is indicated symbolically by "symbol."

Remedy:

The user should attempt to discover why the indicated quantity was zero. Usually potential zero divisors were supposed to be anticipated during the analysis leading to the coding of the CAS Simulator. The program logic should avoid the calculation of zero divisors.

2. Message:

IF NT = 1, VARIANCE ERRORS AND CONFIDENCE LEVELS CAN NOT BE COMPUTED AND WILL ARBITRARILY SET TO ZERO.

Meaning:

Non-fatal error -- on the first Monte Carlo iteration it is not possible to compute the variance errors VEA_C, VEP_C, and VEY_C and the confidence levels CLWA, CLPRD, etc. These values will arbitrarily set to zero.

Remedy:

Not required.

3. Message:

STARTING REGION r_{st} NOT FOUND ON filename FILE.
(where filename is YESOUT, SUBHST, or CAMSF)

Meaning:

Fatal error -- the starting region r_{st} specified by STARTR on the CAS control cards was not found on the indicated file. Either STARTR is incorrect or something is wrong with the indicated file.

Remedy:

Change STARTR or mount the correct file. It might be necessary to dump part of the file to determine the starting region and zone numbers.

4. Message:

STARTING ZONE z_{st} NOT FOUND ON filename FILE
(where filename is YESOUT, SUBHST, or CAMSF)

Meaning:

Fatal error -- the starting zone z_{st} specified by STARTZ on the CAS control cards was not found on the indicated file. Either STARTZ is incorrect or something is wrong with the indicated file.

Remedy:

Change STARTZ or mount the correct file. It might be necessary to dump part of the file to determine the starting region and zone numbers.

5. Message:

ENDING REGION r_{end} NOT FOUND ON filename FILE
(where filename is YESOUT, SUBHST, or CAMSF)

Meaning:

Non-fatal error -- the ending region r_{end} specified by ENDR on the CAS control cards was not found on the indicated file. Either ENDR is incorrect or something is wrong with the indicated file. The program will use all regions up to the end of data on the file.

Remedy:

Change ENDR. Zero is a permissible value indicating to use all regions up to the end of data.

6. Message:

ENDING ZONE z_{end} NOT FOUND ON filename FILE
(where filename is YESOUT, SUBHST, or CAMSF)

Meaning:

Non-fatal error -- the ending zone z_{end} specified by ENDZ on the CAS control cards was not found on the indicated file. Either ENDZ is incorrect or something is wrong with the indicated file. The program will use all zones up to the end of the last region or the region indicated by ENDR.

Remedy:

Change ENDZ. Zero is a permissible value indicating to use all zones of the final region (ENDR).

7. Message:

ZERO PREDICTION DATES ON YESOUT FILE FOR REGION r, ZONE z, STRATUM s (DATA RECORD n)

Meaning:

Fatal error -- all six prediction dates from the YESOUT file are zero for the indicated region, zone, and stratum. Thus the program cannot determine which value of estimated yield to use. Something must be wrong with the YESOUT file.

Remedy:

Dump out part of the YESOUT file to check the prediction dates and yields. In particular record n+1 should be checked.

8. Message:

ILLEGAL GROUP NUMBER g FROM SUBHST FOR REGION r, ZONE z, STRATUM s, SUBSTRATUM k (DATA RECORD n)

Meaning:

Fatal error -- the group number g from the n^{th} data record of the Substrata Historical File SUBHST is not 1, 2, or 3, indicating something is wrong with the SUBHST file.

Remedy:

Dump out the n+1 record of file SUBHST to check the group number and other substrata data.

9. Message:

NAGR = n OR NA = m FROM FILE SUBHST ARE ZERO. GROUP NUMBER g IS CHANGED TO 3.

Meaning:

Non-fatal error -- one or both of the quantities NAGR and NA from the Substrata Historical File are zero. Thus the group number g was changed to 3 by the CAS Simulator.

Remedy:

The input to the LUMP program, which generated SUBHST. If NAGR or NA are zero, then the group number should be 3.

10. Message:

INCONSISTENCY BETWEEN YESOUT AND SUBHST FILES.

	RECORD	REGION	ZONE	STRATA
YESOUT	n_1	r_1	z_1	s_1
SUBHST	n_2	r_2	z_2	s_2

Meaning:

Fatal error -- the region, zone, and strata from the YESOUT and SUBHST files do not agree. Agreement was supposed to be assured by the YES module.

Remedy:

This error should never occur in production. The logic of the YES and CAS modules should be carefully checked. Also it might be necessary to dump portions of the two files.

11. Message:

INCONSISTENCY BETWEEN SUBHST AND CAMSF.

	RECORD	REGION	ZONE	STRATA	SUBSTRATA
SUBHST	n_1	r_1	z_1	s_1	k_1
CAMSF	n_2	r_2	z_2	s_2	k_2

Meaning:

Fatal error -- the region, zone, strata, and substrata from the SUBHST and CAMSF files do not agree. Agreement should have been assured by the CAMS module. Actually, the CAMS module uses the CROPW file rather than SUBHST, but the two files should agree with each other and thus with CAMSF.

Remedy:

This error should never occur in production. The logic of the CAMS and CAS modules should be carefully checked. Also, it might be necessary to dump portions of the two files.

12. Message:

ERROR RETURN FROM BETA DISTRIBUTION SUBROUTINE.
ERROR FLAG = n.

Meaning:

Non-fatal or fatal error, depending upon error flag (see writeup of BETAD routine) -- an error was detected by the BETAD subroutine while CAS was attempting to compute PW'_K, the production wheat for the most recent non-epoch year (eq. (13) in CAS).

The meaning of the error flag is as follows:

1. $\overline{X}$ not within range $0 \le \overline{X} \le 1$ so was reset within BETAD.

2. σ not within range

$$0 \le \sigma \le \overline{X} \sqrt{\frac{1 - \overline{X}}{\overline{X} + \epsilon}}$$

where $\epsilon = 10^{-4}$

so σ was reset within BETAD.

3. Fatal error. The random number could not be found within 35 iterations via the inverse incomplete Beta function method.

$\overline{X}$ is the mean value $\tilde{PW}_K$

σ is the standard deviation

$$\sigma = CV_1 * \tilde{PW}_K$$

13. Message:

NO SEGMENTS IN SUBSTRATA k, STRATA s, ZONE z, REGION r (SUBHST RECORD n).

Meaning:

Fatal error -- the program detected a group I substrata with no segments. Only group II or group III substrata with no segments are permitted.

Remedy:

Check record n+1 of the SUBHST file (in particular check GRPNO and NSEG). This error should never occur during production.

14. Message:

ZERO OR NEGATIVE DIVISOR IN COMPUTING TAU2A, SIGM2S (EQS. 93D-93F)

Meaning:

Fatal error -- the denominator DENOM = HWA12 + RN2(v) * $(MYV12)^{1/2}$ in Eqns. 93d, 93e, and 93f in the calculation of τ_S^2 and σ_S^2 is zero or negative (subroutine DSIO). This probably indicates that the group I, II historical wheat area and the multiyear variance are zero. The program logic should never reach this point (see message 15).

Remedy:

Modify the input to the LUMP program so the historical wheat area is non-zero.

15. Message:

WARNING... HIST PW = $\tilde{PW}$ FOR SUBSTRATA k, STRATA s,
ZONE z, REGION r.
GROUP NO. CHANGED TO 3.

Meaning:

Non-fatal -- the program will not accept a group I or group II sub-strata with a zero (or negative) value of historical PW from the SUBHST file (see eqns. 16, 33, 39, 46, 90, 93).

Remedy:

Non required -- the program will automatically change the group number to 3 and proceed. The user may wish to enter a non-zero value of HIST PW in the LUMP input data.

16. Message:

INPUT PREDICTION DATE (m) = d
.LT. ALL PREDICTION DATES ON YESOUT FILE FOR STRATA s,
ZONE z, REGION r (RECORD n).

Meaning:

Non-fatal -- the m^{th} Zulu prediction date (obtained from the m^{th} prediction date on the CAS input control card data) is less than all prediction dates on the YESOUT file for the indicated stratum on the n^{th} YESOUT data record).

Remedy:

The error is non-fatal. The program will drop the indicated stratum and proceed. However, the user may wish to check the prediction dates entered on the CAS control cards.

17. Message:

TOO MANY MONTE CARLO ITERATIONS FOR THE CAS DISTRIBUTION FILE.

Meaning:

Fatal error -- a maximum of 100 Monte Carlo iterations is allowed, if the CAS distribution file is to be generated.

Remedy:

Specify NTRIAL ≤ 100 in the LEM control card data or specify DISTFF = 0 in the CAS control card data. If more than 100 iterations are required and if the CAS distribution file is desired, then the dimensions of the arrays CASDSB and BUFFR in common block /CASCNM/ may have to be increased. Also the routine RWDISF would have to be modified.

18. Message:

SYMBOL IN EQ. n = a
REF. VALUE = b

Meaning:

Non-fatal -- in subroutine YSUB, which calculates a quantity Y, the argument a for the square root is negative, which could cause trouble. To avoid the problem, the program resets

$$a = 0. \quad \text{if} \quad a < 0$$

and prints a warning if

$$|a| \geq b \times 10^{-7}$$

Remedy:

Non required -- the error is non-fatal, and execution will continue with $a = 0$ and $Y = 10^{-30}$. However, if $|a|$ is significantly large, the user should investigate why.

Note: To prevent excessive amounts of printout, this message will be printed a maximum of five times per iteration.

19. Message:

LESS THAN 2 ACQUIRED SEGMENTS IN CLASS c OF ZONE z REGION r
NO. OF ACQ. SEGMENTS = n

Meaning:

Warning -- There was only one acquired segment in the indicated class c of zone z, region r. Hence, neither the regression formula nor the variance formula could be used to compute S^2, the within-class estimated area variance. S^2 will be set to zero.

Remedy:

This is a non-fatal error and there probably is not much the user can do about the situation. Execution will proceed with S^2 set to zero for the indicated class.

20. Message:

EITHER TOO MANY SUBSTRATA OR SEGMENTS IN REGION-NNNN
ZONE-NNNN
FATAL ERRORS IN PASS 0 OF CAS. RUN ABORTED.

Meaning:

Fatal error -- A maximum of 300 substrata or 300 acquired segments are allowed in a zone.

Remedy:

Reallocate the segments so that there are not so many in any one zone or increase threshold values for acquisition.

APPENDIX A

NOMENCLATURE

A.1 Introduction

In Section A.3 of this Appendix are listed most of the engineering symbols used in the CAS Problem Description. In order to shorten the list, only the primary forms of many of the symbols are given. The conventions described below may be used to distinguish between related forms of the same basic symbol.

Let v be an arbitrary quantity (e.g., WA for wheat area). Then

v denotes the true value;

$\tilde{v}$ denotes the historical value;

$\hat{v}$ denotes the estimated value;

v_ℓ denotes the value for level ℓ

where

$\ell = i$ for a segment,
$\ell = K$ for a substrata,
$\ell = S$ for a stratum,
$\ell = Z$ for a zone,
$\ell = R$ for a region,
$\ell = C$ for a country;

$v_{1\ell}$ denotes the value of v for Group I segments aggregated to level ℓ;

$v_{2\ell}$ denotes the value of v for Group II segments aggregated to level ℓ;

$v_{3\ell}$ denotes the value of v for Group III substrata aggregated to level ℓ.

Thus, for example,

$\hat{WA}_{2S}$ denotes the estimated wheat area for Group II segments aggregated up to the strata level,

$\tilde{PW}_{K}$. denotes the historical proportion of wheat at the substrata level,

$\hat{PRD}_{Z}$ denotes the estimated production of wheat at the zone level.

A.2 Summation Notation

The summation notation Σ is used extensively in the CAS Problem Description to indicate aggregation of various quantities up to certain levels.

Thus,

$\sum_{i}^{K}$ denotes aggregation of segments up to the substrata level,

$\left.\begin{array}{l} \sum^{S} \\ \sum_{i,K}^{S} \end{array}\right\}$ denote aggregation up to the strata level,

$\left.\begin{array}{l} \sum^{Z} \\ \sum_{i,K}^{Z} \end{array}\right\}$ denote aggregation up to the zone level,

$\sum^{R}$ denotes aggregation up to the region level,

$\sum^{C}$ denotes aggregation up to the country level,

$\sum_{i}^{M1K}$ denotes aggregation of Group I segments up to the substrata level,

$\sum_{i}^{M2K}$ denotes aggregation of Group II segments up to the substrata level,

$\sum_{i,K}^{S1}$ denotes aggregation of Group I segments up to the strata level,

$\sum_{K}^{S1}$ denotes aggregation of Group I substrata up to the strata level,

$\sum_{i,K}^{S2}$ denotes aggregation of Group II segments up to the strata level,

$\sum_{K}^{S2}$ denotes aggregation of Group II substrata up to the strata level,

$\sum_{K}^{S3}$ denotes aggregation of Group III substrata up to the strata level,

$\sum^{NT}$ denotes accumulation over all Monte Carlo iterations,

$\sum_{i,K}^{class}$ denotes summation over all substrata in a class.

A.3 Definition of Engineering Symbols

	Symbol	Data Set(s)	Description
1.	A	—	Regression coefficient used to calculate S^2.
2.	$(AREA)_K$	(SUBHST)	Land area of the K^{th} substratum.
3.	AREAPS	(Block Data)	Area per segment. (Built-in value = 10289.712 hectares)
4.	B	—	Regression coefficient used to calculate S^2.
5.	CL $\hat{W}$A	(13)	Confidence level about the estimated WA.
6.	CL WA (True/Est.)	(13)	Confidence level about the true WA using the estimated variance.
7.	CL WA (True/WC)	(13)	Confidence level about the true WA using the within county variance.
8.	CL P$\hat{R}$D	(13)	Confidence level about the estimated production.
9.	CL PRD (True/Est.)	(13)	Confidence level about the true production using the estimated variance.
10.	CL PRD (True/WC)	(13)	Confidence level about the true production using the within county variance.

	Symbol	Data Set(s)	Description
11.	$CT_{1\ell}$	(1)-(6) and (10)-(17)	Number of Group I substrata.
	$CT_{2\ell}$		Number of Group II substrata.
	$CT_{3\ell}$		Number of Group III substrata.
12.	CV_1	(SUBHST)	Coefficient of variation for year-to-year change in PW.
13.	CV_2	(SUBHST)	Coefficient of variation for within county variation of PW.
14.	CV_3	(")	Coefficient of variation for within county variation of proportion of mixed pixels.
15.	CV_4	(")	Ratio of 1964 Historical WA to (1969) Historical WA (used to compute T).
16.	E_ℓ	(7) - (10)	Ratio of estimated group 1, 2 WA to historical group 1, 2 WA.
17.	EA_ℓ	(10)-(17)	Error in WA.
18.	EP_ℓ	(10)-(17)	Error in production.
19.	EY_ℓ	(10)-(17)	Percent error in yield.
20.	H	(Input)	Minimum number of segments required for applying S^2 regression equation.
21.	M	(Input)	Number of historical years for Group III ratio calculation. (No longer used)
22.	M_{1j}	(4)	Number of Group I segments at strata level.
23.	$M_{1\ell}$	(1)-(17)	Number of Group I segments which have been acquired.
24.	M_{2j}	(5)	Number of Group II segments at strata level.
25.	$M_{2\ell}$	(1)-(17)	Number of Group II segments which have been acquired.
26.	NA	(SUBHST)	Number of allocated segments in the substratum (from Substrata Historical File).

Symbol	Data Set(s)	Description
27. N_K	(SUBHST)	Number of agricultural segments (NAGR from Substrata Historical File).
28. NT	—	Number of Monte Carlo iterations.
29. PRD_ℓ	(10)-(17)	Production.
30. PW	—	Proportion wheat.
31. PW_ℓ	(1)-(7)	Proportion wheat.
32. $PW_{1\ell}$	(1), (4)	Proportion of wheat for Group I segments.
33. $PW_{2\ell}$	(2), (5)	Proportion of wheat for Group II segments.
34. $(PW_K)_i$	—	Proportion wheat for segment i and substratum K.
35. P(X)	—	Analytic function used to compute confidence levels.
36. Q	—	Intermediate quantity used to calculate $\hat{V}_{2S}$.
37. R_K	—	Intermediate quantity used to calculate $\hat{WA}_K$.
38. RN	—	Random number used to compute WA'_K.
39. $RN1(\nu)$	—	Random number used to compute τ^2. (No longer used)
40. RN2(v)	—	Random number used to compute τ^2. (No longer used)
41. S^2	(7)	Intermediate factor used to compute $\hat{V}_{1S}$ and $\hat{V}_{2S}$ (within county area variance).
42. S_o^2	—	Computed value of S^2.
43. T	(5)	Intermediate factor used to compute $\hat{V}_{2S}$.
44. $\hat{V}_{1\ell}$	—	Estimated Group I area variance.
45. $\hat{V}_{2\ell}$	—	Estimated Group II area variance.

Note $\hat{V}_{1\ell} + \hat{V}_{2\ell} = \sum_{S=1}^{\ell} (\hat{V}_{1S} + \hat{V}_{2S})$.

	Symbol	Data Set(s)	Description
46.	$V\hat{A}R_{\ell}$	(10)-(17)	Area variance.
47.	$V\hat{P}R_{\ell}$	(10)-(17)	Production variance.
48.	$V\hat{Y}R_{\ell}$	(10)-(17)	Yield variance.
49.	WA	(1)-(17)	Wheat area.
50.	WA_{ℓ}	(1)-(17)	Wheat area.
51.	$WA_{1\ell}$	(1), (4)	Wheat area from Group I segments.
52.	$WA_{2\ell}$	(2), (5)	Wheat area from Group II segments.
53.	$WA_{3\ell}$	(3), (6)	Wheat area from Group III segments.
54.	WA'_{K}	(2)	Most recent non-epoch year.
55.	$(WA_{1,2})_{\ell}$	(7)-(10)	Combined wheat area from Group I and Group II segments.
56.	Y_{ℓ}	(10)-(17)	Yield.
57.	Y_S	(10)	True yield for strata (from YES Output File).
58.	ν	—	Year index used in computing σ^2 and τ^2. (No longer used)
59.	π_{α} π_K $\pi_{K'}$	—	Intermediate quantities used to compute T. π_{α}, π_K, $\pi_{K'}$ all represent the same set of quantities. The index α, K, or K' is a dummy index used to distinguish different substrata.
60.	$\pi_{K''}$	—	Another intermediate quantity used to compute T. $\pi_{K'}$ is computed by a different equation than π_{α}, π_K, $\pi_{K'}$.
61.	σ	—	Standard deviation used in Beta distribution.
62.	σ_{ℓ}^2	—	Intermediate quantity used to compute $V\hat{A}R_S$. (No longer used)
63.	τ_{ℓ}^2	—	Intermediate quantity used to compute $V\hat{A}R_S$.

APPENDIX B

DATA SET DEFINITIONS

The equations within each data set are listed in approximate order of computation.

Data Set 1 (Group I Substrata)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	ISUBST			Substrata ID
2.	TWAK	WA_{1K}	(7)	True WA (wheat area)
3.	HWAK	$\widetilde{WA}_{1K}$	(8)	Historical WA
4.	EWAK	$\hat{WA}_{1K}$	(9)	Estimated WA
5.	EPWK	$\hat{PW}_{1K}$	(2a)	Sum of estimated PW_{Ki} (proportion wheat)
6.	EPW2K		(3a)	$\sum_i \hat{PW}_{1i}^2$
7.	M1K	M_{1K}		No. of Group I segments which have been acquired
8.	SMPKPI		(4a)	$\sum_i (\widetilde{PW}_K)(\hat{PW}_{1i})$
9.	SUMPK2		(5a)	$\sum_i \widetilde{PW}_K^2$
10.	SUMPK		(6a)	$\sum_i \widetilde{PW}_K$
11.	CT1K	CT_1		Group I flag: = 1 if any acquired segments in substrata, = 0 otherwise
12.	ANALVK		(12)	Group I analytic variance
13.	NCLASS			Substrata class number

Data Set 2 (Group II Substrata)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	ISUBST			Substrata ID
2.	TWAK	WA_{2K}	(7)	True WA (wheat area)
3.	HWAK	$\widetilde{WA}_{2K}$	(8)	Historical WA
4.	NEYWAK	WA'_K	(14)	Non-epoch year WA (No longer used)
5.	EPWK	$\hat{PW}_{2K}$	(2b)	Estimated PW (proportion wheat)
6.	EPW2K		(3b)	$\sum_i \hat{PW}_{2i}^2$
7.	M2K	M_{2K}		No. of Group II segments which have been acquired
8.	SMPKPI		(4b)	$\sum_i (\widetilde{PW}_K)(\hat{PW}_{2i})$
9.	SUMPK2		(5b)	$\sum_i \widetilde{PW}_K^2$
10.	SUMPK		(6b)	$\sum_i \widetilde{PW}_K$
11.	CT2K	CT_2		Group II flag: = 1 if any acquired segments in substrata, = 0 otherwise
12.	ANALVK		(17)	Group II Analytic variance
13.	NCLASS			Substrata class number

Data Set 3 (Group III Substrata)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	ISUBST			Substrata ID
2.	TWAK	WA_{3K}	(7)	True WA (wheat area)
3.	HWAK	$\widetilde{WA}_{3K}$	(8)	Historical WA
4.	CT3K	CT_3		Group III flag: = 1 if substrata is Group III or reclassified as Group III

Data Set 4 (Group I Component of Strata Data)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	STRATA			Strata ID
2.	TWAS1	WA_{1S}	(19)	True WA (wheat area)
3.	HWAS1	$\widetilde{WA}_{1S}$	(20)	Historical WA
4.	EWAS1	$\widehat{WA}_{1S}$	(21)	Estimated WA
5.	M1JS	M_{1j}	(22)	Number of acquired Group I segments in strata
6.	CT1S		(37)	No. of Group I substrata with acquired segments
7.	ANVS1		(25)	Group I analytic variance

Data Set 5 (Group II Component of Strata Data)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	STRATA			Strata ID
2.	TWAS2	WA_{2S}	(31)	True WA (wheat area)
3.	HWAS2	$\widetilde{WA}_{2S}$	(32)	Historical WA
4.	EWAS2	$\hat{WA}_{2S}$	(33)	Estimated WA
5.	M2JS	M_{2j}	(34)	Number of acquired Group II segments in strata
6.	CT2S		(38)	No. of Group II substrata with acquired segments
7.	T	T	(39)	Second term in variance equation for $\hat{V}_{2S}$
8.	ANVS2		(40)	Group II analytic variance
9.	P2IDPK		(46)	$\sum_{i,K}^{S2} \frac{\hat{PW}_{2i}}{(\widetilde{PW}_K)_i}$

Data Set 6 (Group III Component of Strata Data)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	STRATA			Strata ID
2.	TWAS3	WA_{3S}	(47)	True WA (wheat area)
3.	HWAS3	$\widetilde{WA}_{3S}$	(48)	Historical WA
4.	CT3S			No. of Group III substrata

In addition to Data Sets 4, 5, and 6, the following strata-dependent quantities need to be written on an intermediate scratch file for second pass processing:

YS = Y_S = True yield for strata

ESTYS = $\hat{Y}_S$ = Estimated yield for strata

EVYRS = $V\hat{Y}R_S$ = Variance of yield for strata

V1V2S = $V_{1S}+V_{2S}$ = Estimated group 1, 2 area variance

VARS = V_S = Estimated area variance for strata

ANVARS = Estimated analytic area variance for strata

Data Set 7 (At Zone Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	ZONE			Zone ID
2.	TWAZ	WA_Z	(56)	True WA (wheat area) (Also in Data Set 11)
3.	HWAZ2	$(\tilde{WA}_{1,2})_Z$	(57)	Group 1, 2 historical WA (if M1K2KZ ≥ 2)
4.	EZ	E_Z	(63)	
5.	M1K2KZ		(49)	$\sum_K^Z (M_{1K} + M_{2K})$
6.	ANALVZ		(69)	Group 1, 2 analytic area variance
7.	NSTRAZ			No. of strata in zone
8.	HWAZ1	$(\tilde{WA}_{1,2})_Z$	(61)	Group 1, 2 historical WA (if M1K2KZ ≥ 1)
9.	EWAZ1	$(\hat{WA}_{1,2})_Z$	(62)	Group 1, 2 estimated WA (if M1K2KZ ≥ 1)
10.	HWAZ3	$\tilde{WA}_{3Z}$	(68)	Total historical wheat area for all strata in zone without acquired segments
11.	ESTVZ		(59)	Group 1, 2 variance estimate
12.	HWAZ12	$(\tilde{WA}_{1,2})_\ell$	(93a)	Group 1, 2 historical WA • Obtained from zone level if M1K2KZ ≥ 2; • HWAR12 at region level if M1K2KZ < 2
13.	M1K2CL		(50)	$\sum_K^{class} (M_{1K} + M_{2K})$ = number of acquired segments in class
14.	EPWCL		(51)	$\sum_{i,K}^{class} (\hat{PW}_{1i} + \hat{PW}_{2i})$
15.	EPW2CL		(52)	$\sum_{i,K}^{class} (\hat{PW}_{1i}^2 + \hat{PW}_{2i}^2)$

	Program Symbol	Engineer Symbol	Equation	Definition
17.	PKPICL		(53)	$\sum_{i,K}^{class} \tilde{PW}_K (\hat{PW}_{1i} + \hat{PW}_{2i})$
18.	PK2CL		(54)	$\sum_{i,K}^{class} \tilde{PW}_K^2 = \sum_{K}^{class} (M_{1K} + M_{2K}) \tilde{PW}_K^2$
19.	PKCL		(55)	$\sum_{i,K}^{class} \tilde{PW}_K = \sum_{K}^{class} (M_{1K} + M_{2K}) \tilde{PW}_K$

Data Set 8 (At Region Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	REGION			Region ID
2.	TWAR	WA_R	(70)	True WA (wheat area) (also in Data Set 12)
3.	HWAR2	$(\tilde{WA}_{1,2})_R$	(71)	Group 1, 2 historical WA
4.	ER	E_R	(77)	
5.	M1K2KR		(78)	$\sum_{K}^{R} (M_{1K} + M_{2K})$
6.	ANALVR		(79)	Group 1, 2 analytic area variance
7.	NZONES			No. of zones in region
8.	HWAR1	$(\tilde{WA}_{1,2})_R$	(75)	Group 1, 2 historical WA
9.	EWAR1	$(\hat{WA}_{1,2})_R$	(76)	Group 1, 2 estimated WA
10.	ESTVR		(73)	Group 1, 2 estimated variance
11.	M1M2ZR			Group 1, 2 flag: = 0 if M1K2KZ < 2 for all zones in region = 1 if M1K2KZ ≥ 2 for any zone in region (i.e., if any zone has at least two group 1, 2 segments)

Data Set 9 (At Country Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	COUNTR			Country ID
2.	TWAC	WA_C	(80)	True WA (wheat area) (Also in Data Set 13)
3.	HWAC2	$(\widetilde{WA}_{1,2})_C$	(81)	Group 1, 2 historical WA
4.	EC	E_C	(87)	
5.	M1K2ZC		(88)	$\sum_K^C (M_{1K} + M_{2K})$
6.	ANALVC		(89)	Group 1, 2 analytic area variance
7.	M1M2FC			Group 1, 2 flag: = 0 if M1K2KZ < 2 for all zones in country = 1 if M1K2KZ ≥ 2 for any zone in country (i.e., if any zone has at least two group 1, 2 segments)
8.	HWAC1	$(\widetilde{WA}_{1,2})_C$	(85)	Group 1, 2 historical WA
9.	EWAC1	$(\widehat{WA}_{1,2})_C$	(86)	Group 1, 2 estimated WA
10.	ESTVC		(83)	Group 1, 2 estimated variance

Data Set 10 (At Strata Level - Part 2 Processing)
Note: Perform only if M1K2KC ≥ 2

	Program Symbol	Engineer Symbol	Equation	Definition
1.	HWAS	$\tilde{WA}_S$	(94)	Historical WA
2.	TWAS	WA_S	(95)	True WA
3.	EWAS	$\hat{WA}_S$	(96)	Estimated WA
4.	AERRS	EA_S	(97)	Area error
5.	AVARS	$\hat{VAR}_S$	(92b, 93c)	Area variance
6.	TPRODS	PRD_S	(99)	True production
7.	EPRODS	$\hat{PRD}_S$	(100)	Estimated production
8.	PRERRS	EP_S	(101)	Production error
9.	PRVARS	$\hat{VPR}_S$	(102)	Production variance
10.	YERRS	EY_S	(103)	Yield error
11.	ANAVS		(92c, 93d)	Analytic area variance
12.	ANPRVS		(105)	Analytic production variance
13.	ES	E_S	(90)	

Data Set 11 (At Zone Level - Part 2 Processing)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	HWAZ	$\widetilde{WA}_Z$	(106)	Historical WA
2.	TWAZ	WA_Z	(107)	True WA
3.	EWAZ	$\hat{WA}_Z$	(108)	Estimated WA
4.	AERRZ	EA_Z	(109)	Area error
5.	AVARZ	$V\hat{A}R_Z$	(110)	Area variance
6.	TPRODZ	PRD_Z	(111)	True production
7.	EPRODZ	$P\hat{R}D_Z$	(112)	Estimated production
8.	PRERRZ	EP_Z	(113)	Production error
9.	PRVARZ	$V\hat{P}R_Z$	(114)	Production variance
10.	TYZ	Y_Z	(115)	True yield
11.	EYZ	$\hat{Y}_Z$	(116)	Estimated yield
12.	YERRZ	EY_Z	(117)	Yield error
13.	M1Z	M_{1Z}	(118)	Number of acquired Group I segments in zone
14.	M2Z	M_{2Z}	(119)	Number of acquired Group II segments in zone
15.	CT1Z	CT_{1Z}	(120)	Number of Group I substrata with acquired segments
16.	CT2Z	CT_{2Z}	(121)	Number of Group II substrata with acquired segments
17.	CT3Z	CT_{3Z}	(122)	Number of Group III substrata with acquired segments
18.	ANAVZ		(123)	Analytic area variance
19.	ANPRVZ		(124)	Analytic production variance

Data Set 12 (At Region Level - Part 2 Processing)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	HWAR	$\widetilde{WA}_R$	(125)	Historical WA
2.	TWAR	WA_R	(126)	True WA
3.	EWAR	$\widehat{WA}_R$	(127)	Estimated WA
4.	AERRR	EA_R	(128)	Area error
5.	AVARR	$V\hat{A}R_R$	(129)	Area variance
6.	TPRODR	PRD_R	(130)	True production
7.	EPRODR	$P\hat{R}D_R$	(131)	Estimated production
8.	PRERRR	EP_R	(132)	Production error
9.	PRVARR	$V\hat{P}R_R$	(133)	Production variance
10.	TYR	Y_R	(134)	True yield
11.	EYR	$\hat{Y}_R$	(135)	Estimated yield
12.	YERRR	EY_R	(136)	Yield error
13.	M1R	M_{1R}	(137)	Number of acquired Group I segments in region
14.	M2R	M_{2R}	(138)	Number of acquired Group II segments in region
15.	CT1R	CT_{1R}	(139)	Number of Group I substrata with acquired segments
16.	CT2R	CT_{2R}	(140)	Number of Group II substrata with acquired segments
17.	CT3R	CT_{3R}	(141)	Number of Group III substrata with acquired segments
18.	ANAVR		(142)	Analytic area variance
19.	ANPRVR		(143)	Analytic production variance

Data Set 13 (At Country Level - Part 2 Processing)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	HWAC	$\tilde{WA}_C$	(144)	Historical WA
2.	TWAC	WA_C	(145)	True WA
3.	EWAC	$\hat{WA}_C$	(146)	Estimated WA
4.	AERRC	EA_C	(147)	Area error
5.	AVARC	$\hat{VAR}_C$	(148)	Area variance
6.	TPRODC	PRD_C	(149)	True production
7.	EPRODC	$\hat{PRD}_C$	(150)	Estimated production
8.	PRERRC	EP_C	(151)	Production error
9.	PRVARC	$\hat{VPR}_C$	(152)	Production variance
10.	TYC	Y_C	(153)	True yield
11.	EYC	$\hat{Y}_C$	(154)	Estimated yield
12.	YERRC	EY_C	(155)	Yield error
13.	M1C	M_{1C}	(156)	Number of acquired Group I segments in country
14.	M2C	M_{2C}	(157)	Number of acquired Group II segments in country
15.	CT1C	CT_{1C}	(158)	Number of Group I substrata with acquired segments
16.	CT2C	CT_{2C}	(159)	Number of Group II substrata with acquired segments
17.	CT3C	CT_{3C}	(160)	Number of Group III substrata with acquired segments
18.	ANAVC		(161)	Analytic area variance
19.	ANPRVC		(162)	Analytic production variance

	Program Symbol	Engineer Symbol	Equation	Definition
20.	CLEWA	CL $\hat{WA}$	(163)	Confidence level about estimated WA using estimated variance
21.	CLEPRD	CL $\hat{PRD}$	(165)	Confidence level about estimated production using estimated variance
22.	CLATEC	CL WA (True/est)	(166)	Confidence level about true WA using estimated variance
23.	CLPTEC	CL PRD (True/est)	(167)	Confidence level about true production using estimated variance
24.	CLATWC	CL WA (True/WC)	(168)	Confidence level about true WA using within county variance
25.	CLPTWC	CL PRD (True/WC)	(169)	Confidence level about true production using within county variance

Data Sets 14, 15, 16, and 17 are similar to Data Sets 10, 11, 12 and 13 with a few important exceptions:

1) Data Sets 10-13 are for only one Monte Carlo iteration, whereas Data Sets 14-17 represent the sums of the values accumulated over all Monte Carlo iterations. These accumulated values will be used to compute and print out average values.

2) Data Sets 14-17 are written onto the CAS Cumulative File.

3) The sums of the squares of the errors are added to Data Sets 14-17.

Thus the additional quantities for Data Sets 14-17 are as follows:

Data Set 14 (At Strata Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	SQAERS	$\sum^{NT} (EA_S)^2$	(170)	Sum of the squares of the area errors
2.	SQPERS	$\sum^{NT} (EP_S)^2$	(171)	Sum of the squares of the production errors
3.	SQYERS	$\sum^{NT} (EY_S)^2$	(172)	Sum of the squares of the yield errors

Data Set 15 (At Zone Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	SQAERZ	$\sum^{NT} (EA_Z)^2$	(173)	Sum of the squares of the area errors
2.	SQPERZ	$\sum^{NT} (EP_Z)^2$	(174)	Sum of the squares of the production errors
3.	SQYERZ	$\sum^{NT} (EY_Z)^2$	(175)	Sum of the squares of the yield errors

Data Set 16 (At Region Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	SQAERR	$\sum^{NT} (EA_R)^2$	(176)	Sum of the squares of the area errors
2.	SQPERR	$\sum^{NT} (EP_R)^2$	(177)	Sum of the squares of the production errors
3.	SQYERR	$\sum^{NT} (EY_R)^2$	(178)	Sum of the squares of the yield errors

Data Set 17 (At Country Level)

	Program Symbol	Engineer Symbol	Equation	Definition
1.	SQAERC	$\sum^{NT} (EA_C)^2$	(179)	Sum of squares of the area errors
2.	SQPERC	$\sum^{NT} (EP_C)^2$	(180)	Sum of the squares of the production errors
3.	SQYERC	$\sum^{NT} (EY_C)^2$	(181)	Sum of the squares of the yield errors

Data Set 18 (At Country Level)

This data set is computed only after the last Monte Carlo trial.

	Program Symbol	Engineer Symbol	Equation	Definition
1.	CLWA	CL WA	(182)	Confidence level about the true WA
2.	CLPRD	CL PRD	(183)	Confidence level about the true production

APPENDIX C

IMPLEMENTATION EQUATIONS

(1) $$R_K = \frac{(AREA)_K}{NA * (AREAPS)}$$

where

$(AREA)_K$ = Substrata land area in hectares,

NA = No. of allocated segments in a substrata,

AREAPS = Area per segment in hectares.

(2) Summation of Estimated PW

(2a) $$\hat{PW}_{1K} = \sum_{i}^{M1K} \hat{PW}_{1i}$$

(2b) $$\hat{PW}_{2K} = \sum_{i}^{M2K} \hat{PW}_{2i}$$

(3a) $$EPW2K = \sum_{i}^{M1K} \hat{PW}_{1i}^{2}$$

(3b) $$EPW2K = \sum_{i}^{M2K} \hat{PW}_{2i}^{2}$$

(4a) $$SMPKPI = \sum_{i}^{M1K} (\tilde{PW}_K)(\hat{PW}_{1i}) = \tilde{PW}_K \hat{PW}_{1K}$$

(4b) $$SMPKPI = \tilde{PW}_K \hat{PW}_{2K}$$

(5a) $$SUMPK2 = \sum_{i}^{M1K} \tilde{PW}_K^{2} = M_{1K} \tilde{PW}_K^{2}$$

(5b) $$SUMPK2 = \sum_{i}^{M2K} \tilde{PW}_K^{2} = M_{2K} \tilde{PW}_K^{2}$$

(6a) $$SUMPK = \sum_{i}^{M1K} \tilde{PW}_K = M_{1K} \tilde{PW}_K$$

(6b) $\text{SUMPK} = \sum_{i}^{M2K} \widetilde{PW}_K = M_{2K}\,\widetilde{PW}_K$

(7) True wheat area

$$WA_K = N_K * R_K * (\text{AREAPS}) * PW_K$$

where $WA_K = WA_{1K},\ WA_{2K},\ \text{or } WA_{3K}$

$PW_K = PW_{1K},\ PW_{2K},\ \text{or } PW_{3K}$

(8) Historical wheat area

$$\widetilde{WA}_K = N_K * R_K * (\text{AREAPS}) * \widetilde{PW}_K$$

where $\widetilde{WA}_K = \widetilde{WA}_{1K},\ \widetilde{WA}_{2K},\ \text{or } \widetilde{WA}_{3K}$

$\widetilde{PW}_K = \widetilde{PW}_{1K},\ \widetilde{PW}_{2K},\ \text{or } \widetilde{PW}_{3K}$

(9) Estimated wheat area

$$\hat{WA}_{1K} = N_K * R_K * (\text{AREAPS}) * \frac{\hat{PW}_{1K}}{M_{1K}}$$

(10) $\text{MYVK} = (\widetilde{WA}_{1K} * CV_4)^2$ (no longer used)

(11) $\text{VMULTK} = \left(1 - \frac{M_{1K}}{N_K}\right) R_K^{\ 2}\ \frac{N_K^{\ 2}}{M_{1K}}\ (\text{AREAPS})^2$

(12) $\text{ANALVK} = (PW_{1K} * CV_2)^2 * \text{VMULTK}$

(13) $PW'_K = f\,(RN,\ CV_1,\ \widetilde{PW}_K)$ (no longer used)

is computed by the Beta Distribution subroutine given mean $\widetilde{PW}_K$ and $\sigma = \widetilde{PW}_K * CV_1$

(14) $WA'_K = CV4 * \widetilde{WA}_K$

(15) $\text{MYVK} = (\widetilde{WA}_{2K} * CV_4)^2$ (no longer used)

(16) $\text{VMULTK} = \frac{R_K^{\ 2}\ (N_K^{\ 2} - N_K)}{\widetilde{WA}_{2K}} * (\text{AREAPS})^2$ (incomplete - see Eqn. (36))

(17) $\text{ANALVK} = (PW_{2K} * CV_2)^2 * \text{VMULTK} * \widetilde{WA}_{2K}$

(18) $MYVK = (\tilde{W}A_{3K} * CV_4)^2$ (no longer used)

(19) $TWAS1 = \sum_{K}^{S1} WA_{1K}$

(20) $HWAS1 = \sum_{K}^{S1} \tilde{W}A_{1K}$

(21) $EWAS1 = \sum_{K}^{S1} \hat{W}A_{1K}$

(22) $M1JS = \sum_{K}^{S1} M_{1K}$

(23) $MYVS1 = \sum_{K}^{S1} MYVK$ (no longer used)

(24) $VMULS1 = \sum_{K}^{S1} VMULTK$ (no longer used)

(25) $ANVS1 = \sum_{K}^{S1} ANALVK$

(26) $EPWS1 = \sum_{i,K}^{S1} P\hat{W}_{1i} = \sum_{K}^{S1} EPWK$ (no longer used)

(27) $EPW2S1 = \sum_{i,K}^{S1} P\hat{W}_{1i}^{\,2} = \sum_{K}^{S1} EPW2K$ (no longer used)

(28) $PKPIS1 = \sum_{i,K}^{S1} (P\tilde{W}_K)\,(P\hat{W}_{1i}) = \sum_{K}^{S1} SMPKPI$ (no longer used)

(29) $PK2S1 = \sum_{i,K}^{S1} (P\tilde{W}_K)^2 = \sum_{K}^{S1} SUMPK2$ (no longer used)

(30) $PKS1 = \sum_{i,K}^{S1} P\tilde{W}_K = \sum_{K}^{S1} SUMPK$ (no longer used)

(31) $$\text{TWAS2} = \sum_{K}^{S2} WA_{2K}$$

(32) $$\text{HWAS2} = \sum_{K}^{S2} \widetilde{WA}_{2K}$$

(33) $$\text{EWAS2} = \frac{\widetilde{WA}_{2S}}{M_{2j}} \sum_{i,K}^{S2} \frac{\hat{PW}_i}{\widetilde{PW}_K} = \frac{\widetilde{WA}_{2S}}{M_{2j}} * \text{P2IDPK}$$

(34) $$\text{M2JS} = \sum_{K}^{S2} M_{2K}$$

(35) $$\text{MYVS2} = \sum_{K}^{S2} \text{MYVK} \quad \text{(no longer used)}$$

(36) $$\text{VMULTK} = \frac{\widetilde{WA}_{2S}}{M_{2j}} * \text{VMULTK} \quad \text{(see Eqn. (16))}$$

(37) $$\text{CT1S} = \sum_{K}^{S1} CT_1$$

(38) $$\text{CT2S} = \sum_{K}^{S2} CT_2$$

(39) $$T = \begin{cases} 0 \quad \text{(at option of user or if M2JS} < 2) \\ \qquad\qquad \text{or} \\ \displaystyle\sum_{K=1}^{S2-1} \sum_{K'=K+1}^{S2} (\pi_K \, \pi_{K'} - \pi_{K''}) \left(\frac{WA'_K}{\pi_K} - \frac{WA'_{K'}}{\pi_{K'}} \right)^2 \quad \text{(if M2JS} > 1) \end{cases}$$

where

$$\pi_K = M_{2j} \left[\frac{\widetilde{WA}_{2K}}{\widetilde{WA}_{2S}} \right]$$

$$\pi_{K'} = M_{2j} \left[\frac{\widetilde{WA}_{2K'}}{\widetilde{WA}_{2S}} \right]$$

$$\pi_{K''} = \frac{(M_{2j} - 1)}{M_{2j}} \pi_K \pi_{K'} + \frac{M_{2j} - 1}{M_{2j}^2} (\pi_K^2 \pi_{K'} + \pi_K \pi_{K'}^2)$$

$$- \frac{(M_{2j} - 1)}{M_{2j}^3} \pi_K \pi_{K'} \sum_{\alpha=1}^{S2} \pi_\alpha^2 + \frac{2(M_{2j} - 1)}{M_{2j}^3} (\pi_K^3 \pi_{K'} + \pi_K \pi_{K'}^3 + \pi_K^2 \pi_{K'}^2)$$

$$- \frac{3(M_{2j} - 1)}{M_{2j}^4} (\pi_K^2 \pi_{K'} + \pi_K \pi_{K'}^2) \sum_{\alpha=1}^{S2} \pi_\alpha^2 + \frac{3(M_{2j} - 1)}{M_{2j}^5} (\pi_K \pi_{K'}) \left(\sum_{\alpha=1}^{S2} \pi_\alpha^2\right)^2$$

$$- \frac{2(M_{2j} - 1)}{M_{2j}^4} (\pi_K) (\pi_{K'}) \sum_{\alpha=1}^{S2} \pi_\alpha^3$$

$$\pi_\alpha = M_{2j} \left[\frac{\tilde{WA}_{2\alpha}}{\tilde{WA}_{2S}} \right]$$

(40) $$ANVS2 = \left[\frac{\sum_{K}^{S2} ANALVK}{M_{2j}} \right]$$

(41) $$EPWS2 = \sum_{i,K}^{S2} \hat{PW}_{2i} = \sum_{K}^{S2} EPWK$$ (no longer used)

(42) $$EPW2S2 = \sum_{i,K}^{S2} \hat{PW}_{2i}^2 = \sum_{K}^{S2} EPW2K$$ (no longer used)

(43) $$PKPIS2 = \sum_{i,K}^{S2} (\tilde{PW}_K)(\hat{PW}_{2i}) = \sum_{K}^{S2} SMPKPI$$ (no longer used)

(44) $$PK2S2 = \sum_{i,K}^{S2} (\tilde{PW}_K)^2 = \sum_{K}^{S2} SUMPK2$$ (no longer used)

(45) $$PKS2 = \sum_{i,K}^{S2} \tilde{PW}_K = \sum_{K}^{S2} SUMPK$$ (no longer used)

(46) $$\text{P2IDPK} = \sum_{i,K}^{S2} \frac{\hat{PW}_{2i}}{(\tilde{PW}_K)_i} = \sum_{K}^{S2} \frac{1}{\tilde{PW}_K} \sum_{i}^{K} \hat{PW}_{2i}$$

$$= \sum_{K}^{S2} \frac{\hat{PW}_{2K}}{\tilde{PW}_K}$$

(47) $$WA_{3S} = \sum_{K}^{S3} WA_{3K}$$

(48) $$\tilde{WA}_{3S} = \sum_{K}^{S3} \tilde{WA}_{3K}$$

(49) $$\text{M1K2KZ} = \sum^{Z} (\text{M1JS} + \text{M2JS})$$

(50) $$\text{M1K2CL} = \sum^{\text{class}} (\text{M1K} + \text{M2K})$$

(51) $$\text{EPWCL} = \sum^{\text{class}} \text{EPWK}$$

(52) $$\text{EPW2CL} = \sum^{\text{class}} \text{EPW2K}$$

(53) $$\text{PKPICL} = \sum^{\text{class}} \text{SMPKPI}$$

(54) $$\text{PK2CL} = \sum^{\text{class}} \text{SUMPK2}$$

(55) $$\text{PKCL} = \sum^{\text{class}} \text{SUMPK}$$

(56) $$\text{TWAZ} = \sum^{Z} (WA_{1S} + WA_{2S} + WA_{3S})$$

(57) $$\text{HWAZ2} = \begin{cases} 0 & \text{if } \text{M1K2KZ} < 2, \\ \sum^{Z} (\tilde{WA}_{1S} + \tilde{WA}_{2S}) & \text{if } \text{M1K2KZ} \geq 2 \end{cases}$$

(58) $$\text{MYVZ} = \begin{cases} 0 & \text{if } \text{M1K2KZ} < 2, \\ \sum^{Z} (\text{MYVS1} + \text{MYVS2}) & \text{if } \text{M1K2KZ} \geq 2 \end{cases}$$ (no longer used)

(59) $$\text{ESTVZ} = \begin{cases} 0 & \text{if } \text{M1K2KZ} < 2, \\ [\sum^{Z} (V_{1S} + V_{2S} + T) & \text{if } \text{M1K2KZ} \geq 2 \end{cases}$$

(60) $$\text{EWAZ2} = \begin{cases} 0 & \text{if } \text{M1K2KZ} < 2, \\ \sum^{Z} (\hat{WA}_{1S} + \hat{WA}_{2S}) & \text{if } \text{M1K2KZ} \geq 2 \end{cases}$$ (no longer used)

(61) $$\text{HWAZ1} = \begin{cases} 0 & \text{if } \text{M1K2KZ} = 0, \\ \sum^{Z} (\tilde{WA}_{1S} + \tilde{WA}_{2S}) & \text{if } \text{M1K2KZ} \geq 1 \end{cases}$$

(62) $$\text{EWAZ1} = \begin{cases} 0 & \text{if } \text{M1K2KZ} = 0, \\ \sum^{Z} (\hat{WA}_{1S} + \hat{WA}_{2S}) & \text{if } \text{M1K2KZ} \geq 1 \end{cases}$$

(63) $$\text{EZ} = \begin{cases} 0 & \text{if } \text{M1K2KZ} = 0, \\ \dfrac{\text{EWAZ1}}{\text{HWAZ1}} & \text{if } \text{M1K2KZ} \geq 1 \end{cases}$$

(64) $$B = \frac{\text{M1K2CL} * \text{PKPICL} - \text{EPWCL} * \text{PKCL}}{\text{M1K2CL} * \text{PK2CL} - \text{PKCL}^2}$$

(65) $$A = \frac{\text{EPWCL} - B * \text{PKCL}}{\text{M1K2CL}}$$

(66) $$S_o^2 = \frac{\text{EPW2CL} - \dfrac{\text{EPWCL}^2}{\text{M1K2CL}}}{\text{M1K2CL} - 1} \quad \text{if } 2 \leq \text{M1K2CL} < H$$

$$S_o^2 = \frac{\text{EPW2CL} - A * \text{EPWCL} - B * \text{PKPICL}}{\text{M1K2CL} - 2} \quad \text{if } \text{M1K2KZ} \geq H$$

S_o^2 is not defined for M1K2CL < 2.

(67) $S^2 = \mathrm{MIN}\left(\frac{(\mathrm{AREAPS})^2}{4}, S_o^{\ 2}\right)$

S^2 is not defined for M1K2CL < 2.

(68) $\mathrm{HWAZ3} = \sum^{Z} (\mathrm{HWAS1} + \mathrm{HWAS2} + \mathrm{HWAS3})$ over all strata without acquired segments

(69) $\mathrm{ANALVZ} = \begin{cases} 0 & \text{if } \mathrm{M1K2KZ} < 2 \\ \sum^{Z} (\mathrm{ANVS1} + \mathrm{ANVS2} + \mathrm{T}) & \text{if } \mathrm{M1K2KZ} \geq 2 \end{cases}$

(70) $\mathrm{TWAR} = \sum^{R} \mathrm{TWAZ}$

(71) $\mathrm{HWAR2} = \sum^{R} \mathrm{HWAZ2}$

(72) $\mathrm{MYVR} = \sum^{R} \mathrm{MYVZ}$ (no longer used)

(73) $\mathrm{ESTVR} = \sum^{R} \mathrm{ESTVZ}$

(74) $\mathrm{EWAR2} = \sum^{R} \mathrm{EWAZ2}$ (no longer used)

(75) $\mathrm{HWAR1} = \sum^{R} \mathrm{HWAZ1}$

(76) $\mathrm{EWAR1} = \sum^{R} \mathrm{EWAZ1}$

(77) $E_R = \frac{\mathrm{EWAR1}}{\mathrm{HWAR1}}$

(78) $\mathrm{M1K2KR} = \sum^{R} \mathrm{M1K2KZ}$

(79) $\mathrm{ANALVR} = \sum^{R} \mathrm{ANALVZ}$

(80) $\mathrm{TWAC} = \sum^{C} \mathrm{TWAR}$

(81) $\mathrm{HWAC2} = \sum^{C} \mathrm{HWAR2}$

(82) $\mathrm{MYVC} = \sum^{C} \mathrm{MYVR}$ (no longer used)

(83) $\text{ESTVC} = \sum^{C} \text{ESTVR}$

(84) $\text{EWAC2} = \sum^{C} \text{EWAR2}$ (no longer used)

(85) $\text{HWAC1} = \sum^{C} \text{HWAR1}$

(86) $\text{EWAC1} = \sum^{C} \text{EWAR1}$

(87) $E_C = \dfrac{\text{EWAC1}}{\text{HWAC1}}$

(88) $\text{M1K2KC} = \sum^{C} \text{M1K2KR}$

(89) $\text{ANALVC} = \sum^{C} \text{ANALVR}$

(90) $$E_S = \begin{cases} \dfrac{\text{EWAS1} + \text{EWAS2}}{\text{HWAS1} + \text{HWAS2}} & \text{if } \text{M1JS} + \text{M2JS} \geq 1 \\ E_Z & \text{if } \text{M1JS} + \text{M2JS} = 0 \text{ and } \text{M1K2KZ} \geq 1 \\ E_R & \text{if } \text{M1K2KZ} = 0 \text{ and } \text{M1K2KR} \geq 1 \\ E_C & \text{if } \text{M1K2KR} = 0 \text{ and } \text{M1K2KC} \geq 1 \end{cases}$$

(91) $\text{V1V2S} = \sum_{K} \text{VMULTK} * \text{SSQ (NCLASS)}$

where NCLASS is the class number for substrata K.

The summation is over all substrata with acquired segments.

(92a) $$TAU2S = \left(1 + \frac{HWAS3}{HWAS1 + HWAS2}\right)^2$$

(92b) AVARS = TAU2S * (V1V2S + T) for strata with acquired segments

(92c) ANAVS = TAU2S * (ANVS1 + ANVS2 + T) for strata with acquired segments

(93a) $$HWAZ12 = \begin{cases} HWAZ2 & \text{if } M1K2KZ \geq 2 \\ HWAR2 & \text{if } M1K2KZ < 2 \text{ and } M1M2ZR \neq 0 \\ HWAC2 & \text{if } M1M2ZR = 0 \end{cases}$$

(93b) $$WRATIO = \left(\frac{HWAS}{HWAZ12}\right)^2$$

(93c) AVARS = WRATIO * ESTVZ if M1K2KZ < 2 or
if M1JS + M2JS = 0

(93d) ANAVS = WRATIO * ANALVZ if M1K2KZ < 2 or
if M1JS + M2JS = 0

(94) $HWAS = HWAS1 + HWAS2 + HWAS3$

(95) $TWAS = TWAS1 + TWAS2 + TWAS3$

(96) $EWAS = EWAS1 + EWAS2 + E_S * HWAS3$

(97) $AERRS = EWAS - TWAS$

(98) $AVARS = ESTVS * TAU2S + 2 * (EWAS12)^2 * SIGM2S$ (replaced by Eqns. 92b, 93c)

(99) $TPRODS = YS * TWAS$

(100) $EPRODS = ESTYS * EWAS$

(101) $PRERRS = EPRODS - TPRODS$

(102) $PRVARS = AVARS * (ESTYS^2 - EVYRS) + EWAS^2 * EVYRS$

(103) $YERRS = \left(\frac{ESTYS - YS}{YS}\right) * 100$

(104) $ANAVS = F * TAU2S + 2 * EWAS12^2 * SIGM2S$ (replaced by Eqns. 92c, 93d)

where

$$F = \begin{cases} ANVS1 + ANVS2 + T & \text{if } M1JS + M2JS \geq 1 \text{ and } M1K2KZ \geq 2 \\ ANALVZ & \text{if } M1JS + M2JS = 0 \text{ and } M1K2KZ \geq 2 \\ ANALVR & \text{if } M1K2KZ < 2 \text{ and } M1M2ZR = 1 \\ ANALVC & \text{if } M1M2ZR = 0 \end{cases}$$

(105) $ANPRVS = ANAVS * (ESTYS^2 - EVYRS) + EWAS^2 * EVYRS$

(106) $HWAZ = \sum^{Z} HWAS$

(107) $TWAZ = \sum^{Z} TWAS$

(108) $EWAZ = \sum^{Z} EWAS$

(109) $AERRZ = EWAZ - TWAZ$

(110) $AVARZ = \sum^{Z} \left(V_{1S} + V_{2S} + T\right) * \left(1 + \frac{HWAZ3}{HWAZ12} + \frac{HWAS3}{HWAS1 + HWAS2}\right)^2$

(111) $TPRODZ = \sum^{Z} TPRODS$

(112) $EPRODZ = \sum^{Z} EPRODS$

(113) $PRERRZ = EPRODZ - TPRODZ$

(114) $PRVARZ = \sum^{Z} PRVARS$

(115) $TYZ = \frac{TPRODZ}{TWAZ}$

(116) $EYZ = \frac{EPRODZ}{EWAZ}$

(117) $YERRZ = \left(\frac{EYZ - TYZ}{TYZ}\right) * 100$

(118) $M1Z = \sum^{Z} M1JS$

(119) $M2Z = \sum^{Z} M2JS$

(120) $CT1Z = \sum^{Z} CT1S$

(121) $CT2Z = \sum^{Z} CT2S$

(122) $CT3Z = \sum^{Z} CT3S$

(123) $ANAVZ = \sum^{Z} (ANVS1 + ANVS2 + T) * \left(1 + \frac{HWAZ3}{HWAZ12} + \frac{HWAS3}{HWAS1 + HWAS2}\right)^2$

(124) $ANPRVZ = \sum^{Z} ANPRVS$

(125) $HWAR = \sum^{R} HWAZ$

(126) $TWAR = \sum^{R} TWAZ$

(127) $EWAR = \sum^{R} EWAZ$

(128) $AERRR = EWAR - TWAR$

(129) $\text{AVARR} = \sum^{R} \text{AVARZ}$

(130) $\text{TPRODR} = \sum^{R} \text{TPRODZ}$

(131) $\text{EPRODR} = \sum^{R} \text{EPRODZ}$

(132) $\text{FRERRR} = \text{EPRODR} - \text{TPRODR}$

(133) $\text{PRVARR} = \sum^{R} \text{PRVARZ}$

(134) $\text{TYR} = \dfrac{\text{TPRODR}}{\text{TWAR}}$

(135) $\text{EYR} = \dfrac{\text{EPRODR}}{\text{EWAR}}$

(136) $\text{YERRR} = \left(\dfrac{\text{EYR} - \text{TYR}}{\text{TYR}}\right) * 100$

(137) $\text{M1R} = \sum^{R} \text{M1Z}$

(138) $\text{M2R} = \sum^{R} \text{M2Z}$

(139) $\text{CT1R} = \sum^{R} \text{CT1Z}$

(140) $\text{CT2R} = \sum^{R} \text{CT2Z}$

(141) $\text{CT3R} = \sum^{R} \text{CT3Z}$

(142) $\text{ANAVR} = \sum^{R} \text{ANAVZ}$

(143) $\text{ANPRVR} = \sum^{R} \text{ANPRVR}$

(144) $\text{HWAC} = \sum^{C} \text{HWAR}$

(145) $\text{TWAC} = \sum^{C} \text{TWAR}$

(146) $EWAC = \sum^{C} EWAR$

(147) $AERRC = EWAC - TWAC$

(148) $AVARC = \sum^{C} AVARR$

(149) $TPRODC = \sum^{C} TPRODR$

(150) $EPRODC = \sum^{C} EPRODR$

(151) $PRERRC = EPRODC - TPRODC$

(152) $PRVARC = \sum^{C} PRVARR$

(153) $TYC = \frac{TPRODC}{TWAC}$

(154) $EYC = \frac{EPRODC}{EWAC}$

(155) $YERRC = \left(\frac{EYC - TYC}{TYC}\right) * 100$

(156) $M1C = \sum^{C} M1R$

(157) $M2C = \sum^{C} M2R$

(158) $CT1C = \sum^{C} CT1R$

(159) $CT2C = \sum^{C} CT2R$

(160) $CT3C = \sum^{C} CT3R$

(161) $ANAVC = \sum^{C} ANAVR$

(162) $ANPRVR = \sum^{C} ANPRVR$

(163) $CLEWA = [2 * P(X) - 1] * 100$

where

$$X = \frac{0.1 * EWAC}{Y}$$

$$Y = MAX\ [(AVARC)^{1/2},\ 10^{-30}]$$

$P(X)$ is given by Equation (164).

(164) $$P(X) = 1 - \frac{1}{2}(1 + 0.196854X + 0.115194X^2 + 0.000344X^3 + 0.019527X^4)^{-4}$$

if X is positive.

$$P(X) = \frac{1}{2}(1 + 0.196854|X| + 0.115194|X|^2 + 0.000344|X|^3 + 0.019527|X|^4)^{-4}$$

if X is negative.

(165) $CLEPRD = [2 * P(X) - 1] * 100$

where

$$X = \frac{0.1 * EPRODC}{Y}$$

$$Y = MAX\ [(PRVARC)^{1/2},\ 10^{-30}]$$

$P(X)$ is given by Equation (164).

(166) $CLATEC = [P(X_1) - P(X_2)] * 100$

where

$$X_1 = \frac{EWAC - 0.9 * TWAC}{Y}$$

$$X_2 = \frac{EWAC - 1.1 * TWAC}{Y}$$

$$Y = MAX\ [(AVARC)^{1/2},\ 10^{-30}]$$

$P(X)$ is given by Equation (164).

(167) $CLPTEC = [P(X_1) - P(X_2)] * 100$

where

$$X_1 = \frac{EPRODC - 0.9 * TPRODC}{Y}$$

$$X_2 = \frac{EPRODC - 1.1 * TPRODC}{Y}$$

$$Y = MAX\ [(PRVARC)^{1/2},\ 10^{-30}]$$

P(X) is given by Equation (164).

(168) $CLATWC = [P(X_1) - P(X_2)] * 100$

where

$$X_1 = \frac{EWAC - 0.9 * TWAC}{Y}$$

$$X_2 = \frac{EWAC - 1.1 * TWAC}{Y}$$

$$Y = MAX\ [(ANAVC)^{1/2},\ 10^{-30}]$$

P(X) is given by Equation (164).

(169) $CLPTWC = [P(X_1) - P(X_2)] * 100$

where

$$X_1 = \frac{EPRODC - 0.9 * TPRODC}{Y}$$

$$X_2 = \frac{EPRODC - 1.1 * TPRODC}{Y}$$

$$Y = MAX\ [(ANPRVC)^{1/2},\ 10^{-30}]$$

P(X) is given by Equation (164).

(170) $SQAERS = \sum^{NT} (AERRS)^2$

(171) $SQPERS = \sum^{NT} (PRERRS)^2$

(172) $SQYERS = \sum^{NT} (YERRS)^2$

(173) $$SQAERZ = \sum^{NT} (AERRZ)^2$$

(174) $$SQPERZ = \sum^{NT} (PRERRZ)^2$$

(175) $$SQYERZ = \sum^{NT} (YERRZ)^2$$

(176) $$SQAERR = \sum^{NT} (AERRR)^2$$

(177) $$SQPERR = \sum^{NT} (PRERRR)^2$$

(178) $$SQYERR = \sum^{NT} (YERRR)^2$$

(179) $$SQAERC = \sum^{NT} (AERRC)^2$$

(180) $$SQPERC = \sum^{NT} (PRERRC)^2$$

(181) $$SQYERC = \sum^{NT} (YERRC)^2$$

(182) $$CLWA = [P(X_1) - P(X_2)] * 100$$

where

$$X_1 = \frac{\sum^{NT} EWAC - 0.9 * \sum^{NT} TWAC}{NT * Y}$$

$$X_2 = \frac{\sum^{NT} EWAC - 1.1 * \sum^{NT} TWAC}{NT * Y}$$

$$Y = MAX\,[(VEA_C)^{1/2},\ 10^{-30}]$$

$$VEA_C = \frac{SQAERC - (\sum^{NT} AERRC)^2/NT}{NT - 1}$$

P (X) is given by Equation (164).

(183) $\text{CLPRD} = [P(X_1) - P(X_2)] * 100$

where

$$X_1 = \frac{\sum^{NT} \text{EPRODC} - 0.9 * \sum^{NT} \text{TPRODC}}{NT * Y}$$

$$X_2 = \frac{\sum^{NT} \text{EPRODC} - 1.1 * \sum^{NT} \text{TPRODC}}{NT * Y}$$

$$Y = \text{MAX}\left[(\text{VEP}_C)^{1/2},\ 10^{-30}\right]$$

$$\text{VEP}_C = \frac{\text{SQPERC} - (\sum^{NT} \text{PERRC})^2/NT}{NT - 1}$$

$P(X)$ is given by Equation (164).

(184) $$\begin{bmatrix}\text{CV AREA EST}\\ \text{(PCT TRUE)}\end{bmatrix}_S = \frac{\sqrt{\dfrac{\sum^{NT} \text{AVARS}}{NT}}}{\dfrac{\sum^{NT} \text{TWAS}}{NT}} \times 100$$

(185) $$\begin{bmatrix}\text{AREA}\\ \text{CV ERROR}\\ \text{(PCT TRUE)}\end{bmatrix}_S = \frac{\sqrt{\dfrac{\text{SQAERS} - (\sum^{NT} \text{AERRS})^2/NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TWAS}}{NT}} \times 100$$

(186) $$\begin{bmatrix}\text{YIELD}\\ \text{ST DEV}\\ \text{PCT ERROR}\end{bmatrix}_S = \sqrt{\frac{\text{SQYERS} - (\sum^{NT} \text{YERRS})^2/NT}{NT - 1}}$$

(187) $$\begin{bmatrix}\text{CV PRD EST}\\ \text{(PCT TRUE)}\end{bmatrix}_S = \frac{\sqrt{\dfrac{\sum^{NT} \text{PRVARS}}{NT}}}{\dfrac{\sum^{NT} \text{TPRODS}}{NT}} \times 100$$

(188) $$\begin{bmatrix} \text{PRD} \\ \text{CV ERROR} \\ \text{(PCT TRUE)} \end{bmatrix}_S = \frac{\sqrt{\dfrac{\text{SQPERS} - \left(\sum^{NT} \text{PRERRS}\right)^2 / NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TPRODS}}{NT}} \times 100$$

(189) $$\begin{bmatrix} \text{CV AREA EST} \\ \text{(PCT TRUE)} \end{bmatrix}_Z = \frac{\sqrt{\dfrac{\sum^{NT} \text{AVARZ}}{NT}}}{\dfrac{\sum^{NT} \text{TWAZ}}{NT}} \times 100$$

(190) $$\begin{bmatrix} \text{AREA} \\ \text{CV ERROR} \\ \text{(PCT TRUE)} \end{bmatrix}_Z = \frac{\sqrt{\dfrac{\text{SQAERZ} - \left(\sum^{NT} \text{AERRZ}\right)^2 / NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TWAZ}}{NT}} \times 100$$

(191) $$\begin{bmatrix} \text{YIELD} \\ \text{ST DEV} \\ \text{PCT ERROR} \end{bmatrix}_Z = \sqrt{\frac{\text{SQYERZ} - \left(\sum^{NT} \text{YERRZ}\right)^2 / NT}{NT - 1}}$$

(192) $$\begin{bmatrix} \text{CV PRD EST} \\ \text{(PCT TRUE)} \end{bmatrix}_Z = \frac{\sqrt{\dfrac{\sum^{NT} \text{PRVARZ}}{NT}}}{\dfrac{\sum^{NT} \text{TPRODZ}}{NT}} \times 100$$

(193) $$\begin{bmatrix} \text{PRD} \\ \text{CV ERROR} \\ \text{(PCT TRUE)} \end{bmatrix}_Z = \frac{\sqrt{\dfrac{\text{SQPERZ} - \left(\sum^{NT} \text{PRERRZ}\right)^2 / NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TPRODZ}}{NT}} \times 100$$

$$(194)\quad \begin{bmatrix}\text{CV AREA EST}\\ \text{(PCT TRUE)}\end{bmatrix}_R = \frac{\sqrt{\dfrac{\sum^{NT} AVARR}{NT}}}{\dfrac{\sum^{NT} TWAR}{NT}} \times 100$$

$$(195)\quad \begin{bmatrix}\text{AREA}\\ \text{CV ERROR}\\ \text{(PCT TRUE)}\end{bmatrix}_R = \frac{\sqrt{\dfrac{SQAERR - (\sum^{NT} AERRR)^2/NT}{NT - 1}}}{\dfrac{\sum^{NT} TWAR}{NT}} \times 100$$

$$(196)\quad \begin{bmatrix}\text{YIELD}\\ \text{ST DEV}\\ \text{PCT ERROR}\end{bmatrix}_R = \sqrt{\frac{SQYERR - (\sum^{NT} YERRR)^2/NT}{NT - 1}}$$

$$(197)\quad \begin{bmatrix}\text{CV PRD EST}\\ \text{(PCT TRUE)}\end{bmatrix}_R = \frac{\sqrt{\dfrac{\sum^{NT} PRVARR}{NT}}}{\dfrac{\sum^{NT} TPRODR}{NT}} \times 100$$

$$(198)\quad \begin{bmatrix}\text{PRD}\\ \text{CV ERROR}\\ \text{(PCT TRUE)}\end{bmatrix}_R = \frac{\sqrt{\dfrac{SQPERR - (\sum^{NT} PRERRR)^2/NT}{NT - 1}}}{\dfrac{\sum^{NT} TPRODR}{NT}} \times 100$$

$$(199)\quad \begin{bmatrix}\text{CV AREA EST}\\ \text{(PCT TRUE)}\end{bmatrix}_C = \frac{\sqrt{\dfrac{\sum^{NT} AVARC}{NT}}}{\dfrac{\sum^{NT} TWAC}{NT}} \times 100$$

$$(200)\quad \begin{bmatrix} \text{AREA} \\ \text{CV ERROR} \\ \text{(PCT TRUE)} \end{bmatrix}_C = \frac{\sqrt{\dfrac{\text{SQAERC} - \left(\sum^{NT} \text{AERRC}\right)^2 / NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TWAC}}{NT}} \times 100$$

$$(201)\quad \begin{bmatrix} \text{YIELD} \\ \text{ST DEV} \\ \text{PCT ERROR} \end{bmatrix}_C = \sqrt{\frac{\text{SQYERC} - \left(\sum^{NT} \text{YERRC}\right)^2 / NT}{NT - 1}}$$

$$(202)\quad \begin{bmatrix} \text{CV PRD EST} \\ \text{(PCT TRUE)} \end{bmatrix}_C = \frac{\sqrt{\dfrac{\sum^{NT} \text{PRVARC}}{NT}}}{\dfrac{\sum^{NT} \text{TPRODC}}{NT}} \times 100$$

$$(203)\quad \begin{bmatrix} \text{PRD} \\ \text{CV ERROR} \\ \text{(PCT TRUE)} \end{bmatrix}_C = \frac{\sqrt{\dfrac{\text{SQPERC} - \left(\sum^{NT} \text{PRERRC}\right)^2 / NT}{NT - 1}}}{\dfrac{\sum^{NT} \text{TPRODC}}{NT}} \times 100$$

$$(204)\quad \begin{matrix} \text{CV} \\ \text{ANAL WA} \\ \text{(PCT TRUE)} \end{matrix} = \frac{\sqrt{\dfrac{\sum^{NT} \text{ANAVC}}{NT}}}{\dfrac{\sum^{NT} \text{TWAC}}{NT}} \times 100$$

$$(205)\quad \begin{matrix} \text{CV} \\ \text{ANAL PRD} \\ \text{(PCT TRUE)} \end{matrix} = \frac{\sqrt{\dfrac{\sum^{NT} \text{ANPRVC}}{NT}}}{\dfrac{\sum^{NT} \text{TPRODC}}{NT}} \times 100$$

PART I

PROBLEM DESCRIPTION

FOR THE YES SUBPROGRAM

1.0 SCOPE

1.1 Program Capabilities. The YES model is designed to simulate the yield estimation process of the LACIE System. The model generates the yield estimates at the strata level of from one to six given estimation points in a simulation season. The estimates are computed from the true yield given in the input data, taking into account the effects of various estimation errors. The output from YES is used by CAS in calculating the production estimates. An option allows the estimated yields to be the same as the true yields, bypassing the error simulation. A printed report of the estimated yields is optional.

1.2 Program Development and Organization. This subprogram will be developed in FORTRAN as an overlay of the LEM program. See the LEM problem description, Section 1.2.

1.3 Operational Assumptions. See the LEM problem description, Section 1.3.

2.0 INPUT

There is one input file. The control card input for YES is included on the LEM control card.

2.1 Cards. See the LEM control card Section 2.1. Inputs relevant to YES include:

RSEED5	the initial random number seed for use in simulating the yield error.	
IYES	= 0, 1	estim. yield includes error simulation
	= 3	estim. yield = true yield
IPRYES	= 0	printed report for 1st and last iterations
	= 1	printed report for all iterations
	= 2	printed report for last iteration only
	= 3	no printed report

2.2 Files. The only input file to YES is the YES ERROR MODEL FILE (YESERR), generated by the SEE program. See Section 2.4 of the Users Manual for the format and contents.

3.0 PROCESSING

See Figure 1 for a flow diagram of YES. The heart of YES is the calculation of the equation:

$$YSCI = YSTR + BIAS + RN*SD$$

for each estimation point (up to 6) for each strata, where:

YSCI = estimated yield

YSTR = true yield

BIAS = bias factor

SD = standard deviation

input quantities from YES input file

RN = random number from a normal distribution

4.0 OUTPUT

4.1 Print Data. The printed report is optional, determined by the LEM control card input, IPRYES. See Figure 2 for an example report.

4.2 Files. The only output file from YES is the YES file, used by CAS. See Section 2.4 of the Users Manual for the format and contents.

CASE ______ PAGE ______

YES YIELD ESTIMATE DATA REPORT - ITERATION NO. ____

COUNTRY ______ REGION ______ ZONE ______ STRATUM ______

PREDICTION DATE MO/DY/YR	TRUE YIELD QUIN/HECTAR	ESTIM. YIELD QUIN./HECTAR	PER CENT ERROR	STANDARD. DEV. QUIN./HECTAR
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______

3 STRATA PER PAGE

COUNTRY ______ REGION ______ ZONE ______ STRATUM ______

PREDICTION DATE TRUE YIELD ESTIM. YIELD PERCENT STANDARD DEV.

Figure 2. YES Report Format

5.0 ERROR PROCESSING

There are two possible errors besides system errors in YES, which generate the following messages:

YES INPUT FILE (YESERR) - BEGINNING REGION AND ZONE NOT FOUND

YES INPUT FILE (YESERR) - ENDING REGION AND ZONE NOT FOUND

If the beginning region and zone are not found, this is a fatal error, and causes return of control immediately to LEM. If the ending region and zone are not found, this generates a warning, but LEM will continue, having processed all records from the beginning region and zone to the end of file. The beginning and ending regions and zones are specified on the LEM control cards.

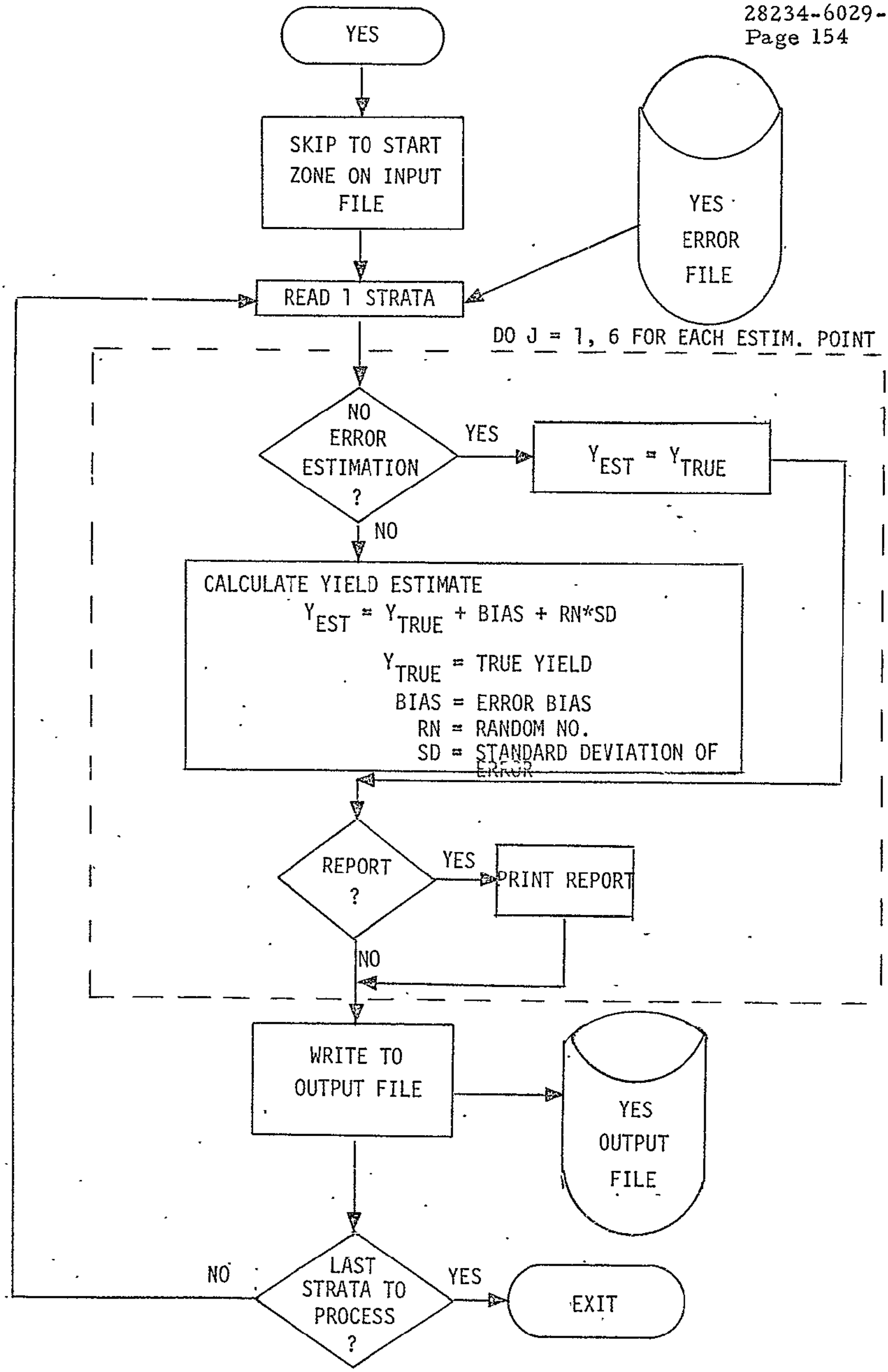


Figure 1. YES Flow Diagram

PART II

LEM COMMON BLOCKS

COMMON BLOCKS FOR THE LEM PROGRAM

COMMON BLOCK	DESCRIPTION
ARGLST	Argument list for error processing
CAMSCM	CAMS control card input data
CASCM	CAS control card input data and constants
CASCUM	Data block for CAS cumulative file
CASFLG	Flags and counters for CAS simulator
CNTRL	Control parameters for LEM program
CONST	Constant quantities for LEM program
DSET1	CAS data sets 1, 2, and 3
DSET4	CAS data sets 4, 5, and 6 (at strata level)
DSET7	CAS data set 7 (at zone level)
DSET8	CAS data set 8 (at region level)
DSET9	CAS data set 9 (at country level)
DSET10	CAS data set 10 (strata data -- second pass)
DSET11	CAS data set 11 (zone data -- second pass)
DSET12	CAS data set 12 (region data -- second pass)
DSET13	CAS data set 13 (country data -- second pass)
FILES	File definitions and record lengths
IXCASF	Index record for CAS cumulative file (CASF)
IXCDSF	Index record for CAS intermediate data set fi .e (CASDSF)
IXDISF	Index record for CAS distribution file
LEMCM	LEM control card input data
PAGECM	Page eject control parameters for LEM
SEGDTA	Segment data from CAMS output file (CAMSF)
SSHDTA	Substrata Historical data from SUBHST file
STATS	Statistical information for LEM
STGDTA	Data for Segment Truth Generator
SUMDTA	Summary data for reports
YESDTA	Yield data from YESOUT file
FILES1	Supplemental file definitions
CLSTAB	Contains data necessary to compute class numbers
IXSUBH	Index record for CAS intermediate SUBHST file

COMMON STORAGE ALLOCATION

Name ARGLST Size 14 Page 1 of 1

Function Argument List for Error Processing

Name	Dimension	Format	Description	Symbol	Units
NERRS	1	I	Number of nonfatal input errors	-	-
NFATAL	1	I	Number of fatal input errors	-	-
NPERRS	1	I	Number of errors during processing	-	-
NARG	1	I	Number of arguments in list	-	-
ARG	10	R	Argument list (real)	-	-
IARG	10	I	Argument list (integer)	-	-
			NOTE: ARG and IARG are equivalenced.		

COMMON STORAGE ALLOCATION

Name CAMSCM Size 138 Page 1 of 1

Function CAMS control input (see also Input, CAMS Problem Description, Section 2.1)

Name	Dimension	Format	Description	Symbol	Units
IMODEL		I1	=1 mixed crops model =2 simple model		
IMULTI		I1	=0 include multi-temporal error ≠0 bypass multi-temporal error		
ISIGEX		I1	=0 additive model } of signature ≠0 multiplicative model } extension		
ISKIP		I1	=0 skip } what to do if no ≠0 classify as training } correlation		
ITMAX		I2	Max. no. days between training/ordinary segment correlation		
IREP		I1	=0 print error breakdown report ≠0 no print of error breakdown report		
IWIND		I1	1-4 which window to use 0 defaults to 4		
			Multi-Temporal Matrix		
IGROUP	(3, 2, 15)	I	Dimension 3 = which M value to use for each of 15 states		
MS	(3, 2, 3)	R	Values for M1, M2, M3		
			Crop Calendar Coefficients		
G	(3, 2, 2)	R	G1 and G2 values for quadratic function		
H	(3, 2, 2)	R	H1 and H2 values for quadratic function		
			Dimension 1 for IGROUP, MS, G, H = TYPE (wheat, mixed, other)		
			Dimension 2 for IGROUP, MS, G, H = SEASON (winter, spring)		
		Note:	If model 2, only wheat dimension of		
			IGROUP, MS, G, H, non-empty		

COMMON STORAGE ALLOCATION

Name CASCM Size 100 Page 1 of 3

Function CAS Control card Input Data and Constants

Name	Dimension	Format	Description	Symbol	Units
AREACF	1	F	Area conversion factor for printout	-	-
			2.471044E-4 (hectares to 10,000 acres)		
			or 0.001 (hectares to 1000 hectares)		
YCF	1	F	Yield conversion factor for printout	-	-
			1.4869664 (quintals/hectare to bushels/acre)		
			or 1.0		
PRDCF	1	F	Production conversion factor for printout	-	-
			3.6743544E-5 (quintals to 100,000 bushels)		
			or 1 E-4 (quintals to 1000 metric tons)		
APRUTS	4,2	4A6	Area units labels for printed reports	-	-
			APRUTS(1,1) - "TEN THOUSAND ACRES"		
			APRUTS(1,2) - "THOUSAND HECTARES"		
PPRUTS	5,2	5A6	Production units labels for printed reports	-	-
			PPRUTS(1,1) - "HUNDRED THOUSAND BUSHELS"		
			PPRUTS(1,2) - "THOUSAND METRIC TONS"		
YPRUTS	3,2	3A6	Yield units labels for printed reports	-	-
			YPRUTS(1,1) - "BUSHELS/ACRE"		
			YPRUTS(1,2) - "QUINTALS/HECTARE"		

COMMON STORAGE ALLOCATION

Name CASCM　　Size ______　　Page 2 of 3

Function CAS Input Data and Constants

Name	Dimension	Format	Description	Symbol	Units
AREAPS	1	F	Area per segment (builtin value = 10289.712)	-	hectares
S2MAX	1	F	Maximum value of S^2	-	$hectares^2$
			(Built in value = $10289.712^2/4$)		
NHISTY	1	I	Number of historical years for group III	M	years
			ratio calculations		
HH	1	I	Minimum number of segments required for	H	
			applying S^2 regression equation		
TØPT	1	I	T - option flag:	-	-
			= 0 to set T = 0, = 1 to calculate T		
			where T is the second term of the variance equation for $\hat{v}_{2S}$		
AUNITS	1	I	Units option: = 1 for metric units -	-	-
			Area in thousands hectares, yield in quintals/hectare; production in thousand metric tons;		
			= 0 for American units -		
			area in ten thousand acres yield in bushels per acres		
			production in hundred thousand bushels		
DISTFF	1	I	CAS distribution file flag:		
			= 0 to generate CAS distribution file,		
			= 1 otherwise		

COMMON STORAGE ALLOCATION

Name CASCM Size ______ Page 3 of 3

Function CAS Input Data and Constants

Name	Dimension	Format	Description	Symbol	Units
BWIND	4	I	Prediction biowindow flags: IWIND(n) = 1 to process biowindow n,	-	-
			= 0 otherwise		
WPRIOR	4	I	Biowindow priorities:	-	-
			List of biowindows in decreasing order		
			of priority		
APREP	1	I	Area and Production Print flag:	-	-
			= 1 to print Area and Production = 0 otherwise		
IPRD	3,14	I	Prediction dates:	-	-
			IPRD(1,n) = year - 1900		
			IPRD(2,n) = month (1-12)		
			IPRD(3,n) = day		
			The prediction dates must be in ascending order.		
			The first zero date terminates the list		
NPDATE	1	I	Number of prediction dates	-	-
PRDATE	14	I	List of prediction dates	-	-
			(in Zulu date format)		

COMMON STORAGE ALLOCATION

Name CASCUM Size 536 Page 1 of 5

Function Data Block for CAS Cumulative File .

Name	Dimension	Format	Description	Symbol	Units
CASCUM	32	F	Block of data for one strata, zone, region	-	-
			or country for a single prediction point		
			(see attached sheets for details of format		
			of each block)		
BUFFR	504	F	Buffer for one data record from the CAS		
			Cumulative File		
			504 = 18 * 28, 18 prediction points		
			28 words/pred. pt.		
ICASC	32	I	Integer name equivalenced to CASCUM		
DSET14	22	F	Data set 14 (strata level)		
DSET15	22	F	Data set 15 (zone level)		
DSET16	22	F	Data set 16 (region level)		
DSET17	28	F	Data set 17 (country level)		
			NOTE: DSET14, DSET15, DSET16, DSET17 all		
			are equivalenced to CASCUM(5).		

COMMON STORAGE ALLOCATION

Name CASCUM Size ______ Page 2 of 5

Function Data Block for CAS Cumulative File

Name	Dimension	Format	Description	Symbol	Units
SQAERS	1	F	Σ (area error)^2 for strata		
SQAERZ	1	F	Σ (area error)^2 for zone		
SQAERR	1	F	Σ (area error)^2 for region		
SQAERC	1	F	Σ (area error)^2 for country		
			SQAERS, SQAERZ, SQAERR, and SQAERC are all equivalenced to CASCUM (24).		
SQPERS	1	F	Σ (production error)^2 for strata		
SQPERZ	1	F	Σ (production error)^2 for zone		
SQPERR	1	F	Σ (production error)^2 for region		
SQPERC	1	F	Σ (production error)^2 for country		
			SQPERS, SQPERZ, SQPERR, and SQPERC are all equivalenced to CASCUM(25).		
SQYERS	1	F	Σ (yield error)^2 for strata		
SQYERZ	1	F	Σ (yield error)^2 for zone		
SQYERR	1	F	Σ (yield error)^2 for region		
SQYERC	1	F	Σ (yield error)^2 for country		
			SQYERS, SQYERZ, SQYERR, and SQYERC are all equivalenced to CASCUM(26).		

COMMON STORAGE ALLOCATION

Name CASCUM Size ______ Page 3 of 5

Function Data Block for CAS Cumulative File

Name	Dimension	Format	Description	Symbol	Units
CASDSB	303	F	Data Block for the CAS Distribution file (Data Set 19)		
ICASD	303	I	Integer name for CASDSB		
HWA2K	60	F	HWA2K(K) specifies the historical WA for the K^{th} group II substrata in the current stratum	$\widetilde{WA}_{2K}$	hectares
WAKNEY	60	F	WAKNEY(K) specifies the non-epoch year WA for the K^{th} group II substrata in the current strata	WA'_{2K}	hectares
PIK	60	F	PIK(K) specifies π_K for the K^{th} group II substrata in the current strata	π_K	-
			NOTE: CASDSB, ICASD, HWA2K are all equivalenced to BUFFR; WAKNEY is equivalenced to CASDSB(61); PIK is equivalenced to CASDSB(121).		

FORMATS OF CAS CUMULATIVE FILE RECORDS

CASCUM	Data Set 14	Data Set 15	Data Set 16	Data Set 17
1	region	region	region	0
2	zone	zone	0	0
3	strata	0	0	0
4	0	nstraz	0	0
5	HWA	(Historical WA)		
6	TWA	(True WA)		
7	EWA	(Estimated WA)		
8	AERR	(area error)		
9	AVAR	(area variance)		
10	TPRØD	(true production)		
11	EPRØD	(estimated production)		
12	PRERR	(production error)		
13	PRVAR	(production variance)		
14	TY	(true yield)		
15	EY	(estimated yield)		
16	YERR	(yield error)		
17	M1	(no. of group I segments)		
18	M2	(no. of group II segments)		
19	CT1	(no. of group I substrata)		
20	CT2	(no. of group II substrata)		
21	CT3	(no. of group III substrata)		
22	ANAV	(analytic area variance)		
23	ANPRV	(analytic production variance)		

CASCUM	Data Set 14	Data Set 15	Data Set 16	Data Set 17
24	SQAERS	SQAERZ	SQAERR	SQAERC
25	SQPERS	SQPERZ	SQPERR	SQPERC
26	SQYERS	SQYERZ	SQYERR	SQYERC
27	-	-	-	CLEWA
28	-	-	-	CLEPRD
29	-	-	-	CLATEC
30	-	-	-	CLPTEC
31	-	-	-	CLATWC
32	-	-	-	CLPTWC

NOTE: The quantities in CASCUM(5) - CASCUM(23) and CASCUM(27) - CASCUM(32) are the accumulated values of the indicated quantities over all iterations.

COMMON STORAGE ALLOCATION

Name CASFLG   Size 40   Page 1 of 3

Function Miscellaneous Data, Flags and Counters for CAS

Name	Dimension	Format	Description	Symbol	Units
H	1	F	Minimum number of segments required for applying S^2 regression equation.	H	-
PPFLG	1	I	Prediction Point Flag = 0 for biowindows = 1 for prediction dates	-	-
NBW	1	I	Number of biowindows (≤4)	-	-
IBW	1	I	Biowindow index (1-4)	-	-
WINDØW	1	I	Window index (1-4)	-	-
IPD	1	I	Prediction Date index (1-14)	-	-
IPP	1	I	Prediction Point index (including both biowindows and prediction dates)	-	-
PPDATE	1	I	Zulu date associated with prediction point	-	-
NREGS	1	I	Number of regions in country	-	-
NZTØT	1	I	Number of zones in country	-	-
NSTRAT	1	I	Number of strata in country	-	-
NYESSK	1	I	Number of records to initially skip on YESØUT file	-	-
NSSHSK	1	I	Number of records to initially skip on SUBHST file	-	-

COMMON STORAGE ALLOCATION

Name CASFLG Size 40 Page 2 of 3

Function Miscellaneous Data, Flags and Counters forCAS

Name	Dimension	Format	Description	Symbol	Units
NCAMSK	1	I	Number of records to initially skip on	-	-
			CAMS file		
NRYES	1	I	Data record count on YESØUT	-	-
NRSSH	1	I	Data record count on SUBHST	-	-
NRCAMS	1	I	Data record count on CAMSF	-	-
ENDC	1	I	End of country flag: $\neq$ 0 if end region, end zone	-	-
ENDREG	1	I	End of region flag: $\neq$ 0 if end of region reached	-	-
ENDZØN	1	I	End of zone flag: $\neq$ 0 if end of zone reached	-	-
IRSTR	1	I	Record number of strata record on CASCUM and CASDSF	-	-
IRZØNE	1	I	Record number of zone record on CASCUM and CASDSF	-	-
IRREG	1	I	Record number of region record on CASCUM and CASDSF	-	-
LDS1	1	I	Length of Data Sets 1, 2, 3	-	-
LDS4	1	I	Length of Data Sets 4, 5, 6	-	-
LDS7	1	I	Length of Data Set 7	-	-
LDS8	1	I	Length of Data Set 8	-	-
LDS9	1	I	Length of Data Set 9	-	-
LDS10	1	I	Length of Data Set 10	-	-
LDS11	1	I	Length of Data Set 11	-	-

COMMON STORAGE ALLOCATION

Name CASFLG Size 40 Page 3 of 3

Function Miscellaneous Data, Flgas and Counters for CAS

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
LDS12	1	I	Length of Data Set 12	-	-
LDS13	1	I	Length of Data Set 13	-	-
LDS14	1	I	Length of Data Set 14	-	-
LDS15	1	I	Length of Data Set 15	-	-
LDS16	1	I	Length of Data Set 16	-	-
LDS17	1	I	Length of Data Set 17	-	-
LRCØN	1	I	No longer used	-	-
LRREG	1	I	No longer used	-	-
LRZØNE	1	I	No longer used	-	-
LRSTR	1	I	No longer used	-	-

COMMON STORAGE ALLOCATION

Name CNTRL Size 16 Page 1 of 1

Function Control Parameters

Name	Dimension	Format	Description	Symbol	Units
PRINTF	1	I	Print flag for modules	-	-
			STG, CAMS, YES, and CAS		
			= 0 to print reports,		
			= 1 otherwise		
NSTART	1	I	Starting iteration number	-	-
			(= RSTART + 1)		
SEED	7	D.P.	Random number seeds for error sources:	-	-
			SEED(1) - Segment Truth Error		
			SEED(2) - Classification Error		
			SEED(3) - Signature Extension Error		
			SEED(4) - Segment crop calendar error		
			SEED(5) - Yield error		
			SEED(6) - CAS Group II error		
			SEED(7) - CAS Group III error		

COMMON STORAGE ALLOCATION

Name CØNST Size 5 Page 1 of 1

Function CONSTANTS

Name	Dimension	Format	Description	Symbol	Units
NTRMX	1	I	Maximum number of Monte Carlo trials in a single run (=100)	-	-
MAXR	1	I	Maximum region number (=999)	-	-
MAXZ	1	I	Maximum zone number (=999)	-	-
IMXSEG	1	I	Maximum number of segments in any substrata (read from header record of Substrata Historical file)	-	-
ENDFIL	1	F	End of file indicator (=4HZZZZ)	-	-
ITSFG	I	I			
MXCLSS	I	I	Maximum number of substrata classes per zone (10)		

COMMON STORAGE ALLOCATION

Name DSET1 Size 14 Page 1 of 2

Function Data Sets 1, 2, 3 (Substrata Data)

Name	Dimension	Format	Description	Symbol	Units
ISUBST	1	I	Substrata ID	-	-
TWAK	1	F	True Wheat Area (WA)	WA_K	hectares
HWAK	1	F	Historical Wheat Area	$\widetilde{WA}_K$	hectares
EWAK	1	F	Estimated Wheat Area	$\widehat{WA}_K$	hectares
M1K	1	F	No. of group I segments in substratum.	M_{1K}	-
CT1K	1	F	Group I flag:	CT_{1K}	-
			= 1 if substrata is in group I,		
			= 0 otherwise		
ANALVK	1	F	Analytic area variance	-	hectares^2
EPWK	1	F	Estimated proportion of wheat	$\widehat{PW}_K$	-
EPW2K	1	F		$\widehat{PW}_K^2$	-
SMPKPI	1	F	$\sum_i (\widetilde{PW}_K) (\widehat{PW}_i)$		
SUMPK2	1	F	$\sum_i (\widetilde{PW}_K)^2 = M_{1K} \widetilde{PW}_K^2$	-	-
SUMPK	1	F	$\sum_i \widetilde{PW}_K = M_{1K} \widetilde{PW}_K^2$	-	-
KSUB	1	F	Index used to count no. of group II	-	-
			substrata in strata.		
NCLASS	1	I	Substrata class number for current prediction point		

COMMON STORAGE ALLOCATION

Name DSET1 Size 14 Page 2 of 2

Function Data Sets 1, 2, 3 (Substrata Data)

Name	Dimension	Format	Description	Symbol	Units
M2K	1	F	No. of group II segments in substrata	M_{2K}	-
			Note: M2K is equivalenced to M1K		
CT2K	1	F	Group II flag:	CT_{2K}	
			= 1 if substrata is in group II, = 0 otherwise		
CT3K	1	F	Group III flag:	CT_{3K}	-
			= 1 if substrata is in group III = 0 otherwise		
			Note: Since CT2K and CT3K are equivalenced		
			to CT1K, a value 1 is also stored for the location CT1K.		
DSET1	14	F	Data Set 1		
DSET2	14	F	Data Set 2		
DSET3	6	F	Data Set 3		
			Note: DSET1, DSET2, and DSET3 are all		
			equivalenced to ISUBST.		

COMMON STORAGE ALLOCATION

Name DSET4 Size 24 Page 1 of 3

Function Data Sets 4, 5, 6 (at Strata level)

Name	Dimension	Format	Description	Symbol	Units
STRATA	1	I	Strata ID	S	-
TWAS1	1	F	True WA (wheat area)	WA_{1S}	ha
HWAS1	1	F	Historical WA (Group I)	$\widetilde{WA}_{1S}$	ha
EWAS1	1	F	Estimated WA (Group I)	$\widehat{WA}_{1S}$	ha
XM1JS	1	F	No. of acquired group I segments in strata	M_{1j}	
XCT1S	1	F	No. of group I substrata with acquired	CT_{1S}	-
			segments		
ANVS1	1	F	Group I Analytic variance	-	ha^2
TWAS2	1	F	True WA for group II segments	WA_{2s}	ha
HWAS2	1	F	Historical WA (Group II)	$\widetilde{WA}_{2S}$	ha
EWAS2	1	F	Estimated WA (Group II)	$\widehat{WA}_{2S}$	ha
XM2JS	1	F	No. of acquired group II segments in strata	M_{2j}	-
XCT2S	1	F	No. of group II substrata with acquired	CT_{2S}	
			segments		

COMMON STORAGE ALLOCATION

Name DSET4 Size 24 Page 2 of 3

Function Data Sets 4, 5, 6 (at Strata level)

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
ANVS2	1	F	Group II Analytic variance	-	ha^2
T	1	F	2nd term in variance eq.	T	ha^2
TWAS3	1	F	True WA for group III segments	WA_{3S}	ha
HWAS3	1	F	Historical WA (group III)	$\widetilde{WA}_{3S}$	ha
XCT3S	1	F	No. of group III substrata	CT_{3S}	-
XYS	1	F	True yield	Y_S	$\frac{\text{Quintals}}{\text{ha}}$
XESTYS	1	F	Estimated yield	$\hat{Y}_S$	$\frac{\text{Quintals}}{\text{ha}}$
EVYRS	1	F	Variance of yield error	$\widehat{VYR}_S$	$\left(\frac{\text{Quintals}}{\text{ha}}\right)^2$
					-
					-
					-
					-
					-

COMMON STORAGE ALLOCATION

Name DSET4 Size 24 Page 3 of 3

Function Data Sets 4, 5, 6 (at Strata level)

Name	Dimension	Format	Description	Symbol	Units
DSET4	24	F	Data Set 4:	-	-
			Note: Data set 4 proper consists of the first 9 quantities in /DSET4/		
DSET5	7	F	Data Set 5:	-	-
			DSET5 is equivalenced to TWAS2	-	-
DSET6	3	F	Data Set 6:	-	-
			DSET6 is equivalenced to TWAS3		
V1V2S	1	F	Group 1, 2 estimated area variance ($= V_{1S} + V_{2S}$)	$V_{1,2S}$	ha^2
VARS	1	F	Estimated area variance	V_{1S}	ha^2
ANVARS	1	F	Estimated analytic area variance		ha^2

COMMON STORAGE ALLOCATION

Name DSET7 Size 81 Page 1 of 2

Function Data Set 7 (at zone level)

Name	Dimension	Format	Description	Symbol	Units
ZONE	1	I	Zone ID	Z	-
HWAZ2	1	F	Historical Wheat area (for group 1, 2 segments)	$\widetilde{WA}$	ha
⋮			Computed only if M1K2KZ≥2 Otherwise = 0		
EZ	1	F		E_Z	
M1K2KZ	1	F	Number of group 1, 2 segments in zone	M_{1Z}	
ANALVZ	1	F	Group 1, 2 Analytic area variance	-	ha^2
NSTRAZ	1	I	Number of strata in zone	-	-
HWAZ1	1	F	Historical wheat area (group 1,2)	$\widetilde{WA}_{1,2}$	ha
EWAZ1	1	F	Estimated wheat area (group 1, 2)	$\widehat{WA}_{1,2}$	ha
HWAZ3	1	F	Total historical wheat are for all strata without valid segments	-	ha
ESTVZ	1	F	Group 1, 2 variance estimate		ha^2
HWAZ12	1	F	Effective group 1, 2 WA for compute area variance for strata without segments	-	ha
M1K2CL	10	F	Number of segments in each substrata class $= \sum_{CLASS} (M_{1K} + M_{2K})$	-	-
EPWCL	10	F	$\sum_{i,K}^{Z} (\widehat{PW}_{1K} + \widehat{PW}_{2K})$ (for each substrata class)	-	-
EPW2CL	10	F	$\sum_{i,K}^{Z} (\widehat{PW}_{1K}^2 + \widehat{PW}_{2K})$ "	-	-
PKPICL	10	F	$\sum_{i,K} PW_K \; (\widehat{PW}_{1K} + \widehat{PW}_{2K})$ "	-	-

COMMON STORAGE ALLOCATION

Name DSET7 Size 81 Page 2 of 2

Function Data Set 7 (at zone level)

Name	Dimension	Format	Description	Symbol	Units
PK2CL	10	F	$\sum_{i,K} \widetilde{PW}_K^2$ (for each substrata class)	-	-
PKCL	10	F	$\sum_{i,K} \widetilde{PW}_K$ "	-	-
SSQ	10	F	S^2 factor in variance equations	S^2	-
			(for each substrata class)		
DSET7	81	F	Data Set 7	-	-
			Note: DSET7 is equivalenced to ZØNE		

4-3

COMMON STORAGE ALLOCATION

Name DSET8 Size 10 Page 1 of 2

Function Data Set 8 (at Region level)

Name	Dimension	Format	Description	Symbol	Units
REGIØN	1	I	Region ID	R	
HWAR2	1	F	Group 1, 2 Historical WA	$\widetilde{WA}_{1,2}$	ha
ER	1	F		E_R	
M1K2KR	1	F	Number of group 1, 2 segments in region	M_{1R}	-
ANALVR	1	F	Analytic area variance	-	ha^2
NZØNES	1	I	Number of zones in region	-	- -
HWAR1	1	F	Historical wheat area (group 1, 2)	$\widetilde{WA}_{1,2}$	ha
EWAR1	1	F	Estimated wheat area (group 1, 2)	$\widehat{WA}_{1,2}$	ha
			Note: HWAR1 and EWAR1 are always computed if there is at least one group I or group II segment in the region. HWAR2 and EWAR2 are computed only if some zone in the region contains at least two group 1, 2 segments.		
ESTVR	1	F	Group 1, 2 variance estimate		ha^2
M1M2ZR	1	I	Group 1, 2 substrata flag:		
			= 1 if M1K2KZ > for any zone in region,		
			= 0 otherwise		
FILL8	1	F	Filler to make a 25 word block for writing onto CASDSF		

COMMON STORAGE ALLOCATION

Name DSET8 Size 10 Page 2 of 2

Function Data Set 8 (at Region level)

Name	Dimension	Format	Description	Symbol	Units
DSET8	10	F	Data Set 8	-	-
			Note: Data Set 8 is equivalenced to REGIØN.		

COMMON STORAGE ALLOCATION

Name DSET9 **Size** 9 **Page** 1 **of** 1

Function Data Set 9 (at Country level)

Name	Dimension	Format	Description	Symbol	Units
COUNTR	1	A6	Country ID	-	-
HWAC2	1	F	Group 1, 2 Historical $\tilde{WA}$	$\tilde{WA}_{1,2}$	ha
EC	1	F		E_C	-
M1K2KC	1	F	Number of group 1, 2 segments in country	M_{1C}	
ANALVC	1	F	Analytic area variance	-	ha^2
M1M2ZC	1	F	Group 1, 2 substrata flag: = 1 if M1K2KZ>1 for any zone in country	-	-
			= 0 otherwise		
HWAC1	1	F	Historical wheat area (Group 1, 2)	$\tilde{WA}_{1,2}$	ha
EWAC1	1	F	Estimated wheat area (Group 1, 2)	$\widehat{WA}_{1,2}$	ha
			Note: HWAC2 = Σ HWAZ2 HWAC1 = Σ HWAZ1 over all		
			EWAC2 = Σ EWAZ2 zones EWAC1 = Σ EWAZ1		
ESTVC	1	F	Group 1, 2 variance estimate	-	ha^2
DSET9	9	F	Data Set 9		
			Note: DSET9 is equivalenced to COUNTR		

COMMON STORAGE ALLOCATION

Name DSET10 **Size** 20 **Page** 1 **of** 2

Function Data Set 10 (Strata Data - Second Pass)

Name	Dimension	Format	Description	Symbol	Units
HWAS	1	F	Historical WA	$\widetilde{WA}_S$	ha
TWAS	1	F	True WA	WA_S	ha
EWAS	1	F	Estimated WA	$\widehat{WA}_S$	ha
AERRS	1	F	Area error	EA_S	ha
AVARS	1	F	Area variance	$\widehat{VAR}_S$	ha^2
TPRODS	1	F	True production	PRD_S	Quintals
EPRODS	1	F	Estimated production	$\widehat{PRD}_S$	Quintals
PRERRS	1	F	Production error	EP_S	Quintals
PRVARS	1	F	Production variance	$\widehat{VPR}_S$	$Quintals^2$
YS	1	F	True yield	Y_S	$\frac{Quintals}{ha}$
ESTYS	1	F	Estimated yield	$\hat{Y}_S$	$\frac{Quintals}{ha}$
YERRS	1	F	Error in yield	EYS	$\frac{Quintals}{ha}$
M1JS	1	F	Number of group I segments in strata	M_{1j}	-

COMMON STORAGE ALLOCATION

Name DSET10 **Size** 20 **Page** 2 **of** 2

Function Data Set 10 (Strata Data - Final Pass)

Name	Dimension	Format	Description	Symbol	Units
M2JS	1	F	Number of group II segments in strata	M_{2j}	
CT1S	1	F	Number of group I substrata in strata	CT_{1S}	
CT2S	1	F	Number of group II substrata in strata	CT_{2S}	
CT3S	1	F	Number of group III substrata in strata	CT_{3S}	
ANAVS	1	F	Analytic area variance		ha^2
ANPRVS	1	F	Analytic production variance		$Quintals^2$
ES	1	F	Ratio of Group 1, 2 Est. WA to Group 1, 2 Hist. WA	E_S	-
DSET10	20	F	Data Set 10	-	-
			Note: DSET10 is equivalenced to HWAS		

COMMON STORAGE ALLOCATION

Name DSET11 **Size** 19 **Page** 1 **of** 2

Function Data Set 11 (Zone Data - Final Pass)

Name	Dimension	Format	Description	Symbol	Units
HWAZ	1	F	Historical WA	$\widetilde{WA}_Z$	ha
TWAZ	1	F	True WA	WA_Z	ha
EWAZ	1	F	Estimated WA	$\widehat{WA}_Z$	ha
AERRZ	1	F	Area error	EA_Z	ha
AVARZ	1	F	Area variance		ha^2
TPRØDZ	1	F	True production	PRD_Z	Quintals
EPRØDZ	1	F	Estimated production	$\widehat{PRD}_Z$	Quintals
PRERRZ	1	F	Production error	EP_Z	Quintals
PRVARZ	1	F	Production variance		$Quintals^2$
TYZ	1	F	True yield	Y_Z	$\frac{Quintals}{ha}$
EYZ	1	F	Estimated yield	$\widehat{Y}_Z$	$\frac{Quintals}{ha}$
YERRZ	1	F	Yield error	EY_Z	$\frac{Quintals}{ha}$
M1Z	1	F	Number of group I segments in zone	M_{1Z}	-
M2Z	1	F	Number of group II segments in zone	M_{2Z}	-
CT1Z	1	F	Number of group I substrata in zone	CT_{1Z}	-
CT2Z	1	F	Number of group II substrata in zone	CT_{2Z}	-
CT3Z	1	F	Number of group III substrata in zone	CT_{3Z}	-
ANAVZ	1	F	Analytic area variance	-	ha^2

COMMON STORAGE ALLOCATION

Name DSET11 Size 19 Page 2 of 2

Function Data Set 11 (Zone Data - Final Pass)

Name	Dimension	Format	Description	Symbol	Units
ANPRVZ	1	F	Analytic production variance	-	$Quintals^2$
DSET11	19	F	Data Set 11		
			Note: DSET11 is equivalenced to HWAZ		

COMMON STORAGE ALLOCATION

Name DSET12 Size 19 Page 1 of 2

Function Data Set 12 (region data-final pass)

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
HWAR	1	F	Historical WA	$\widetilde{WA}_R$	ha
TWAR	1	F	True WA	WA_R	ha
EWAR	1	F	Estimated WA	$\widehat{WA}_R$	ha
AERRR	1	F	Area error	EA_R	ha
AVARR	1	F	Area variance		ha^2
TPRØDR	1	F	True production	PRD_R	Quintals
EPRØDR	1	F	Estimated production	$\widehat{PRD}_R$	Quintals
PRERRR	1	F	Production error	EP_R	Quintals
PRVARR	1	F	Production variance		$Quintals^2$
TYR	1	F	True yield	Y_R	$\frac{Quintals}{ha}$
EYR	1	F	Estimated yield	$\hat{Y}_R$	$\frac{Quintals}{ha}$
YERRR	1	F	Yield error	EY_R	$\frac{Quintals}{ha}$
M1R	1	F	Number of group I segments in region	M_{1R}	-
M2R	1	F	Number of group II segments in region	M_{2R}	-
CT1R	1	F	Number of group I substrata in region	CT_{1R}	-
CT2R	1	F	Number of group II substrata in region	CT_{2R}	
CT3R	1	F	Number of group III substrata in region	CT_{3R}	
ANAVR	1	F	Analytic area variance	-	ha^2

COMMON STORAGE ALLOCATION

Name DSET12 Size 19 Page 2 of 2

Function Data Set 12 (region data-final pass)

Name	Dimension	Format	Description	Symbol	Units
ANPRVR	1	F	Analytic production variance	-	Quintals[2]
DSET12	1	F	Data Set 12	-	-
			Note: DSET12 is equivalenced to HWAR		

COMMON STORAGE ALLOCATION

Name DSET13 Size 25 Page 1 of 2

Function Data Set 13 (country level-final pass)

Name	Dimension	Format	Description	Symbol	Units
HWAC	1	F	Historical WA	$\widetilde{WA}_C$	ha
TWAC	1	F	True WA	WA_C	ha
EWAC	1	F	Estimated WA	$\widehat{WA}_C$	ha
AERRC	1	F	Area error	EA_C	ha
AVARC	1	F	Area variance		ha^2
TPROD	1	F	True production	PRD_C	Quintals
EPRODC	1	F	Estimated production	$\widehat{PRD}_C$	Quintals
PRERRC	1	F	Production error	EP_C	Quintals
PRVARC	1	F	Production variance		$Quintals^2$
TYC	1	F	True yield	Y_C	$\frac{Quintals}{ha}$
EYC	1	F	Estimated yield	$\widehat{Y}_C$	$\frac{Quintals}{ha}$
YERRC	1	F	Yield error	EY_C	$\frac{Quintals}{ha}$
M1C	1	F	No. of group I segments in country	M_{1C}	-
M2C	1	F	No. of group II segments in country	M_{2C}	-
CT1C	1	F	No. of group I substrata in country	CT_{1C}	-
CT2C	1	F	No. of group II substrata in country	CT_{2C}	-
CT3C	1	F	No. of group III substrata in country	CT_{3C}	-
ANAVC	1	F	Analytic area variance		ha^2

COMMON STORAGE ALLOCATION

Name DSET13 Size 25 Page 2 of 2

Function Data Set 13 (country level-final pass)

Name	Dimension	Format	Description	Symbol	Units
ANPRVC	1	F	Analytic production variance		$Quintals^2$
CLEWA	1	F	Confidence level about estimated WA using estimated variance.	$CL\ \widehat{WA}$	-
CLEPRD	1	F	Confidence level about estimated production using estimated variance.	$CL\ \widehat{PRD}$	-
CLATEC	1	F	Confidence level about true WA using estimated variance.	CL WA true/est	-
CLPTEC	1	F	Confidence level about true production using estimated variance.	CL PRD true/est	-
CLATWC	1	F	Confidence level about true WA using within county variance.	CL WA true/WC	-
CLPTWC	1	F	Confidence level about true production using within county variance	CL PRD true/WC	-

COMMON STORAGE ALLOCATION

Name CLSTAB Size 2724 Page _____ of _____

Function Contains tables necessary to determine class sets within a zone.

Name	Dimension	Format	Description	Symbol	Units
ISTRAT	300	I	Contains strata ID for all strata in a zone		
ISBSTR	300	I	Contains substrata ID for all substrata in a zone		
NSCNT	300	I	No. of acquired segments for each substrata		
IGROUP	300	I	Group no. assignment for each substrata		
IDAT1	300	I	Variable usage. Set to AREAK in CLASSN for use by SEGTAB. SEGTAB resets it to		
			first subscript PTR into XORD for each substrata segment set ≡ DAT1 (used by ASSCLS)		
IDAT2	300	I	Variable usage. Set to HISTPW by CLASSN for use by SEGTAB. ASSCLS puts the		
			assigned class number for each substrata ≡ DAT2		
XORD	300	Flt pt	Ratio for each substrata used to determine class	X_i	
IXPT	300	I	Sorted pointers into XORD (in ascending order)		
IBPT	10	I	For each class the beginning subscript in XORD (found indirect via IXPT lookup)		
IEPT	10	I	For each class the ending subscript in XORD (found indirect via IXPT)		
MAXCLS	1	I	Maximum no. of classes allowed ≡ (10-1)		
ICLCNT	1	I	Actual count on number of classes		
IRANK	300	I	Table of gap rankings of sorted X_i for each substrata in a zone	≡ rank	
ISUB1	1	I	Count of no. of substrata in zone		
NACQ	1	I	No. of acquired segments in a zone		

COMMON STORAGE ALLOCATION

Name IXSUBH Size 2 Page ____ of ____

Function Index information for ISUBH2 file

Name	Dimension	Format	Description	Symbol	Units
LIXSSH	1	I	Length of index record for substrata intermediate file (≡3202)		
IXSUBH	1	I	Index record for substrata intermediate file - ISUBH2		

COMMON STORAGE ALLOCATION

Name FILES1 Size 3 Page ____ of ____

Function ____

Name	Dimension	Format	Description	Symbol	Units
ISUBH2	1	I	The unit no. for the intermediate D. Access SUBHST file = 17		
LSUBH2	1	I	Length of record on ISUBH2 file = 39		
MXCLSS	1	I	Maximum no. of classes allowed = 10		

COMMON STORAGE ALLOCATION

Name FILES   Size 30   Page 1 of 2

Function File Definitions and Record Lengths

Name	Dimension	Format	Description	Symbol	Units
SEGID	1	I	Logical file number for segment ID file		
LSEGID	1	I	Record length for segment ID file		
CROPW	1	I	Logical file number for Crop Window file		
LCROPW	1	I	Record length for Crop Window file		
SUBHST	1	I	Logical file number for Substrata Historical file		
LSUBH	1	I	Record length for Substrata Historical file		
ACQUIS	1	I	Logical file number for Data Acquisition file		
LACQ	1	I	Record length for Data Acquisition file		
CAMSF	1	I	Logical file number for CAMS Output file		
LCAMSF	1	I	Record length for CAMS Output file		
CAMERR	1	I	Logical file number for CAMS Error Model file		
LCAMER	1	I	Record length for CAMS Error Model file		
CASF	1	I	Logical file number for CAS Cumulative Output file		
LCASF	1	I	Record length for CAS Cumulative Output file		
YESØUT	1	I	Logical file number for YES Output file		
LYESØ	1	I	Record length for YES Output file		
SIGEXT	1	I	Logical file number for Signature Extension file		
LSIGEX	1	I	Record length for Signature Extension file		

COMMON STORAGE ALLOCATION

Name FILES Size 30 Page 2 of 2

Function File Definitions and Record Lengths

Name	Dimension	Format	Description	Symbol	Units
YESERR	1	I	Logical file number for YES Error Model file		
LYESER	1	I	Record length for YES Error Model file		
SEGTRU	1	I	Logical file number for Segment Truth file		
LSEGTR	1	I	Record length for Segment Truth file		
CASDIS	1	I	Logical file number for CAS Distribution Output file		
LCASD	1	I	Record length for CAS Distribution Output file		
INP	1	I	Logical file number for input file		
ØUTP	1	I	Logical file number for output file		
TACQ	1	I	Logical file number for CAMS scratch file		
LTACQ		I	Record length for CAMS scratch file		
CASDSF	1	I	CAS Intermediate file		
LCASDS		I	Record length for CAS Intermediate file		

COMMON STORAGE ALLOCATION

Name IXCASF Size 389 Page 1 of 1

Function Index Record for CAS Cumulative File (CASF)

Name	Dimension	Format	Description	Symbol	Units
IXCASF	388	I	Index record for CAS Cumulative file (CASF) (needed on CDC 6000 computer)	-	-
LIXCAS	1	I	Length of CASF index record	-	-

COMMON STORAGE ALLOCATION

Name IXDISF Size 113 Page 1 of 1

Function Index Record for CAS Distribution file (CASDIS)

Name	Dimension	Format	Description	Symbol	Units
IXDISF	1136	I	Index record for CASDIS file	-	-
			(Need on CDC 6000 computer)		
LIXDIS	1	I	Length of CASDIS index record	-	-
			Note: The length of IXDISF must be		
			n*63 + 2 where n = number of prediction points		

COMMON STORAGE ALLOCATION

Name IXCDSF　　Size 389　　Page 1 of 1

Function Index Record for CAS Intermediate file

Name	Dimension	Format	Description	Symbol	Units
IXCDSF	388	I	Index record for CAS Intermediate data set	-	-
			file (CASDSF)		
			(Need on CDC 6000 computing system)		
LIXCDS	1	I	Length of index record for CASDSF	-	--

COMMON STORAGE ALLOCATION

Name LEMCM Size 57 Page 1 of 5

Function LEM Control Card Input Data

Name	Dimension	Format	Description	Symbol	Units
TITLE	10	A6	Problem header printed at the top of each page	-	-
ICASE	1	I	Case number	-	-
CUNTRY	1	A6	Country	-	-
NTRIAL	1	I	Number of Monte Carlo iterations at end of run	-	-
RSTART	1	I	=n ≠0 to restart after n Monte Carlo iterations	-	-
IPRINT	1	I	Print flag for segment truth	-	-
			= 0 to print first and last iterations, = 1 to print every iteration,		
			= 2 to print last iteration, = 3 to skip printing.		
STARTR	1	I	Starting region number	-	-
STARTZ	1	I	Starting zone number	-	-
ENDR	1	I	Ending region number	-	-
ENDZ	1	I	Ending zone number	-	-
ISTG	1	I	Segment Truth Generator Error flag:	-	-
			= 0 to vary error, = 1 to hold error constant using first		
			iteration results throughout run, = 2 to hold error constant using a previously ~~generated segment truth file~~,		
			= 3 to eliminate error (error is zero)		

COMMON STORAGE ALLOCATION

Name LEMCM Size 57 Page 2 of 5

Function LEM Control Card Input Data

Name	Dimension	Format	Description	Symbol	Units
ICAMS	1	I	CAMS Error Flag: (Use is similar to use of ISTG described above)	-	-
IYES	1	I	YES Error flag: (Use is similar to use of ISTG described above)	-	-
IACQ	1	I	Segment Acquisition flag	-	-
			= 0 to include segment acquisition conditions, = 1 to eliminate segment acquisition conditions.		
ICLASS	1	I	Classification Error flag:	-	-
			= 0 to vary classification error, = 1 to hold error constant,		
			= 2 if error is zero.		
ISEXT	1	I	Signature Extension error:	-	-
			= 0 to vary error, = 1 to hold error constant,		
			= 2 if error is zero.		
ISCC	1	I	Segment Crop Calendar error:	-	-
			= 0 to vary error, = 1 to hold error constant,		
			= 2 if error is zero.		
ICAS2	1	I	CAS Group II Error flag:	-	-
			= 0 to vary error, = 1 to hold error constant,		
			= 2 if error is zero.		
ICAS3	1	I	CAS Group III Error flag:	-	-
			= 0 to vary error, = 1 to hold error constant, = 2 if error is zero.		

COMMON STORAGE ALLOCATION

Name LEMCM Size 57 Page 3 of 5

Function LEM Control Card Input Data

Name	Dimension	Format	Description	Symbol	Units
IPRCAM	1	I	Print flag for CAMS	-	-
			= 0 to print report on first and last iterations, = 1 to print every iteration,		
			= 2 to print last iteration only, = 3 to suppress printing.		
IPRYES	1	I	Print flag for YES	-	-
			(similar to IPRCAM)		
IPRCAS	1	I	Print flag for CAS	-	-
			(similar to IPRCAM)		

COMMON STORAGE ALLOCATION

Name LEMCM Size 57 Page 4 of 5

Function LEM Control Card Input Data

Name	Dimension	Format	Description	Symbol	Units
ICSESG	1	I	Case number for Segment ID file	-	-
ICSECW	1	I	Case number for Crop Window (Calendar) file	-	-
ICSESH	1	I	Case number for Substrata Historical file	-	-
ICSECE	1	I	Case number for CAMS Error Model file	-	-
ICSEYM	1	I	Case number for YES Data file	-	-
ICSESE	1	I	Case number for Signature Extension file	-	-
ICSEAC	1	I	Case number for Data Acquisition file	-	-
RSEED1	1	DP	Initial random number seed for Segment Truth Error	-	-
RSEED2	1	DP	Initial random no. seed for Classification Error	-	-
RSEED3	1	DP	Initial random no. seed for Signature Ext. Error	-	-
RSEED4	1	DP	Initial random number seed for segment Crop Calendar Error	-	-
RSEED5	1	DP	Initial random no. seed for yield error	-	-
RSEED6	1	DP	Initial random no. seed for CAS Group II Error	-	-
RSEED7	1	DP	Initial random number seed for CAS Group III Error	-	-
RSEED	7	DP	RSEED ≡ RSEED1		

COMMON STORAGE ALLOCATION

Name LEMCM Size 57 Page 5 of 5

Function LEM Control Card Input Data

Name	Dimension	Format	Description	Symbol	Units
ICSEST	1	I	Case number for Segment Truth File		
ICSECØ	1	I	Case number for CAMS Output file		
ICSEYS	1	I	Case number for YES Output file		
ICSECU	1	I	Case number for CAS Cumulative Output file		
ICSECD	1	I	Case number for CAS Distribution Output file		

COMMON STORAGE ALLOCATION

Name PAGECM   Size 14   Page 1 of 1

Function Common Quantities for PAGER

Name	Dimension	Format	Description	Symbol	Units
NPAGE	1	I	Current Page number	-	-
NLINE	1	I	Current line count for the current page		-
MXLINE	1	I	Maximum number of lines per page		
			(built-in value = 40)		
NSTTL	1	I	Subtitle Option		
			= 1 for subtitle		
SUBTTL	10	A6	Subtitle		

COMMON STORAGE ALLOCATION

Name SEGDTA   Size 20   Page 1 of 1

Function Segment Data From CAMS File

Name	Dimension	Format	Description	Symbol	Units
IDSEGT	5	I	Country, region, zone, strata, and		
			substrata ID's for current segment		
ISEG	1	I	Segment ID		
TPWKI	1	F	True PW for segment	PW_{Ki}	-
			(converted to fraction from %)		
ZACDAY	4	I	Zulu Acquisition day for 4 windows		
EPWKI	4	F	Estimated PW for this segment for 4 windows	$\widehat{PW}_{Ki}$	
			(converted to fraction from %)		
ERRPWI	4	F	Error in PW estimate		
ESTPWI	1	F	Estimated PW for this segment for current	$\widehat{PW}_{Ki}$	
			window, ESTPWI = EPWKI(WINDØW)		

COMMON STORAGE ALLOCATION

Name SSHDTA Size 39 Page 1 of 2

Function Substrata Historical Data from SUBHST

Name	Dimension	Format	Description	Symbol	Units
COUN2	1	A6	Country ID	-	-
IREG	1	I	Region ID	R	-
IZONE2	1	I	Zone ID	Z	-
ISTRA2	1	I	Strata ID	S	-
ISUBS2	1	I	Substrata ID	K	-
NSEG	1	I	Number of segments	-	-
IDSEG	1	I	List of sample segments in this substrata	-	-
GRPNØ	1	I	Group number	-	-
HISTPW	1	F	Historical PW for substrata	PW_K	-
			(fraction converted from %)		
AREAK	1	F	Land area of the substrata		ha
			(in ha converted from KM^2)		
PWK	1	F	True PW for substrata	PW_K	
			(fraction converted from %)		
NAGR	1	I	No. of agricultural segments in substrata	N_K	
NA	1	I	No. of allocated segments in substrata	N_A	
DELTPW	1	F	Bias of true PW	-	
DELTPM	1	F	Ratio of true mixed pixels	-	-
CV1	1	F	Coefficient of variation for year to year change in PW	CV_1	

COMMON STORAGE ALLOCATION

Name SSHDTA Size 20 Page 2 of 2

Function Substrata Historical Data from SUBHST

Name	Dimension	Format	Description	Symbol	Units
CV2	1	F	Coefficient of variation for within	CV_2	
			country variation of PW		
CV3	1	F	Coefficient of variation for within	CV_3	
			county variation of proportion of mixed pixels		
CV4	1	F	Ratio of 1964 ref. WA to Historical	CV_4	-
			(1969) WA		
RDSSH	1	I	Read flag for SUBHST	-	-
			= 0 to read SUBHST ≠ 0 otherwise		
SSHDTA	38		Substrata Historical Data	-	-
			Note: SSHDTA is equivalenced to CØUN2		
VMULTK	1	F	Variance multiplier	-	-
CLASS	18	I	Array of class numbers (one per prediction point)	-	-
MXK	1	I	Count on no. of acquired segments (for GROUP 1/2) only	Not Used	

COMMON STORAGE ALLOCATION

Name STATS Size 13 Page 1 of 1

Function Statistical Information for LEM

Name	Dimension	Format	Description	Symbol	Units
ITER	1	I	Current Monte Carlo iteration number	-	-
NSEGTR	1	I	Number of data records written onto the	-	-
			Segment Truth file		
NCAMSR	1	I	Number of data records written onto the	-	-
			CAMS Output file		
NYESR	1	I	Number of data records written onto the YES Output file	-	-
NREC	7	I	Number of data records read from the input files	-	-
			1 = Segment ID file 2 = Crop Window file		
			3 = Substrata Historical file 4 = CAMS Error Model file		
			5 = YES Error Model file 6 = Signature Extension file		
			7 = Data Acquisition file		
NCASCR	1	I	Number of data records written onto the CAS Cumulative file		
NCASDR	1	I	Number of data records written onto the CAS Distribution file		
NT	1	I	Equivalenced to ITER		

COMMON STORAGE ALLOCATION

Name STGDTA Size 643 Page 1 of 3

Function Data for Segment Truth Generator

Name	Dimension	Format	Description	Symbol	Units
COUN	1	A6	Country ID from SEGID file	-	-
IREG	1	I	Region ID from SEGID file	-	-
IZØNE	1	I	Zone ID from SEGID file	-	-
ISTRAT	1	I	Strata ID from SEGID file	-	-
ISUBS	1	I	Substrata ID from SEGID file	-	-
ISEG	1	I	Segment ID from SEGID file	-	-
ITRAIN	1	I	Training Segment Indicator 1 = normal, 0 training	-	-
ITSPRL	6	I	Training Segment Priority List	-	-
SLAT	1	F	Segment latitude	-	radians
SLØNG	1	F	Segment longitude	-	radians
GRIDNØ	1	I	Grid number	-	-
ISW	1	I	Spring/Winter wheat indicator (0 = winter, 1 = spring)	-	-
CØUN2	1	A6	Country ID from SUBHST file	-	-
IREG2	1	I	Region ID from SUBHST file	-	-
IZØNE2	1	I	Aone ID from SUBHST file	-	-
ISTRA2	1	I	Strata ID from SUBHST file	-	-
ISUBS2	1	I	Substrata ID from SUBHST file	-	-
NSEG	1	I	Number of segments in substrata	-	-

COMMON STORAGE ALLOCATION

Name STGDTA Size 643 Page 2 of 3

Function Data for Segment Truth Generator

Name	Dimension	Format	Description	Symbol	Units
IDSEG	150	I	List of sample segments in substrata	-	-
GRPNØ	1	I	Substrata Group number	-	-
HISTPW	1	F	Historical proportion wheat	-	%
AREA	1	F	Substrata land area	A	KM^2
PWK	1	F	True proportion wheat	PW_K	%
NAGR	1	I	Number of agricultural segments in substrata	N_K	-
NA	1	I	Number of allocated segments in substrata	N_A	-
DELTPW	1	F	Bias of true PW	δPW	-
DELTPM	1	F	Ratio of true mixed pixels	δPM	
CV1	1	F	Coefficient of variation for year-to-year change in PW	CV_1	-
CV2	1	F	Coefficient of variation for within county variation of PW	CV_2	-
CV3	1	F	Coefficient of variation for within county variation of PM	CV_3	-
CV4	1	F	Coefficient of variation of multi-year historical WA	CV_4	-
PWKI	1	F	True proportion wheat for segment i	PW_{Ki}	%
PMKI	1	F	True proportion mixed pixels for segment i	MN_{Ki}	%
AVEPW	1	F	Average segment PW for substrata	$\overline{PW}_{Ki}$	%
SUMPW	1	F	Sum of PW_{Ki} for all segments in substrata	-	%
SNBR	1	F	Number of segments in substrata	-	-

COMMON STORAGE ALLOCATION

Name STGDTA Size 643 Page 3 of 3

Function Data for Segment Truth Generator

Name	Dimension	Format	Description	Symbol	Units
PMMEAN	1	F	Mean proportion mixed pixels	$\overline{PM}_K$	%
PW	150	F	PW(i) = proportion wheat for i^{th}	PW_{Ki}	%
PM	150	F	PM(i) = proportion mixed pixels for i^{th}	PM_{Ki}	%
			segment in substrata		
ERRPW	150	F	Error in PW for i^{th} Segment in substrata		%
SIGMA	1	F	Standard deviation used to compute PW_{Ki},	σ	-
			PM_{Ki}		
ERRPWI	1	F	Not used	-	-

COMMON STORAGE ALLOCATION

Name SUMDTA Size 329 Page 1 of 1

Function Summary Data for Reports

Name	Dimension	Format	Description	Symbol	Units
CVAEPT	1	F	CV Area Est. (Pct. True)		-
CVEPTA	1	F	CV Error (Pct. True) for WA		-
SDPER	1	F	St. Dev. (Pct. Error) for yield		See Report
CVPEPT	1	F	CV PRD Est. (Pct. True)		-
CVEPTP	1	F	CV Error (Pct. True) for production		-
CSUMR	18,18	F	Country Summary Data		See Report
			CSUMR(i,n) specifies the i^{th} value for the n^{th} prediction point		
			CSUMR (1,n) =mean EWAC (2,n) = CV Anal. WA (Pct. true)		
			⋮		
			(18,n) = Prod. Cont. level true/WC		

COMMON STORAGE ALLOCATION

Name YESDTA   Size 21   Page 1 of 1

Function Yield Data from YESOUT file

Name	Dimension	Format	Description	Symbol	Units
YSTR	1	F	True yield for stratum	Y_S	Quintals/ha
IZPRDD	6	I	Zulu yield data for up to six prediction points	-	-
YSCI	6	F	Estimated yields for the six prediction points	Y_S	Quintals/ha
VSYCI	6	F	Variances of yield for the six prediction	VYR_S	Quintals/ha
RDYES	1	I	Flag used to control reading YESØUT file:	-	-
			= 0 to read YESØUT,		
			≠ 0 otherwise		
NYESPP	1	I	No. of YES prediction points for one stratum	-	-

ISUBH2 FILE

SUBHST scratch file for CAS. This file is generated from the SUBHST file and augmented with class numbers in pass 0. This file then is used in place of SUBHST in subsequent CAS passes.

Access Method: Direct with fixed length records -- uses FORTRAN V direct access routines.

Status: Temporary, regenerated everytime CAS runs.

Sort: Country, region, zone, strata, then substrata. 3201 records max.

Media: Disk - FASTRAND

Record Length: 39 words

Recommended Blocking Factor: 5

File Size: 124, 839 words

Record Formats:

Detail Record

COUN2	-	See SUBHST file definition
IREG2	-	"
IZON2	-	"
ISTRA2	-	"
ISUBS2	-	"
NSEG	-	"
IDSEG	-	Dummy cell (not used in CAS)
GRPNO	-	See SUBHST file definition
HISTPW	-	"
AREAK	-	"
PWK	-	"
NAGR	-	"
NA	-	"
DELTPW	-	"

Detail Record (cont'd)

DELTPM	-	See SUBHST file definition
CV1	-	"
CV2	-	"
CV3	-	"
CV4	-	"
VMULTK	-	1 word flt. pt., variance multiplier in hectares
CLASS	-	18 word array (integer), class no. assignment for each of up to 18 prediction points, 0-10
MXK	-	1 word integer, count on no. of acquired segments (for group 1/2 only), 0-300

Trailer Record

COUN2	-	Contains 'ZZZZ'.

The remainder of the record contains 38 zeros.

CAMS COMMON BLOCKS

COMMON STORAGE ALLOCATION

Name ACQUIS Size 107 Page 1 of 1

Function Holds 1 record from Data Acquisition File, input file (ACQUIS)

Name	Dimension	Format	Description	Symbol	Units
COUN1		R	Country ID		
IREG1		I	Region ID		
IZONE1			Zone ID		
ISTRA1			Strata ID		
ISUBST1			Substrata ID		
ISEG1			Segment ID, 4 crop windows		
IWIN	4, 25		1st dimension = 4 crop windows		
			2nd dimension = up to 25 Zulu dates/ window in ascending order		
ITOTAL		I	Total no. of accesses		

COMMON STORAGE ALLOCATION

Name CAMERR Size 50 Page 1 of 1

Function Holds 1 record from CAMS Error File, input file (CAMERR)

Name	Dimension	Format	Description	Symbol	Units
COUN2			Country ID		
IREG2			Region ID		
IZONE2			Zone ID		
ISTRA2			Strata ID		
ISUBST2			Substrata ID		
ISEG2			Segment ID		
PW	3, 4		Probability of classif. as wheat,		
			1st dim. = type (given wheat, mixed, other		
			2nd dim. = window		
BERR	3, 4		Bias error, 1st and 2nd dimensions same as above		
SIGERR	3, 4		Std. dev. of error, 1st and 2nd dimensions same as above		

COMMON STORAGE ALLOCATION

Name CAMSF Size 19 Page 1 of 1

Function Holds 1 record for CAMS Output File

Name	Dimension	Format	Description		Symbol	Units
COUN6			Country ID			
IREG6			Region ID			
IZONE6			Zone ID			
ISTRA6			Strata ID			
ISUB6			Substrata ID			
ISEG6			Segment ID			
PTRUE			True proportion of wheat	for each of 4 windows		
IZULU	4		Zulu date of acquisition			
PEST	4		Estimated proportion of wheat			
PERR	4		Percent error in estimate			

COMMON STORAGE ALLOCATION

Name CROPW Size 33 Page 1 of 1

Function Holds 1 record from Crop Calendar File, input file (CROPW)

Name	Dimension	Format	Description	Symbol	Units
COUN3			Country ID		
IREG3			Region ID		
IZONE3			Zone ID		
ISTRA3			Strata ID		
ISUBST3			Substrata ID		
START	2, 4	I	Zulu date, start of window, dim. 1 = season (winter, spring) dim. 2 = window		
END	2, 4	I	Zulu date, end of window		
SD	2		Std. dev. of seg. cal. error 0 $\longrightarrow$ 99		
ERR	2, 5		Δ # days in error		

COMMON STORAGE ALLOCATION

Name ERROR Size 40 Page 1 of 1

Function Values for CAMS report

Name	Dimension	Format	Description	Symbol	Units
TITLE	4		Window title		
IDATE			Acquisition date		
PESTIM			Estimated proportion of wheat		
TOT			Total error		
ALOCAL			Ordinary segment error without signature extension error		
ERTOT	3		Total error		
ERBIAS	3		Total bias error, dimension = type (wheat, mixed, other)		
ERRAND	3		Total random error		
CLTOT	3		Classification error		
CLBIAS	3		Classification bias component		
CLRAND	3		Classification random conponent		
DELTA			Crop calendar error factor		
CROPD			Crop calendar error factor		
Z	3, 2		Signature extension error factors, dimension 2 = Z1 or Z2		
MULT	3		Multi-temporal error factors		
TID			Segment ID of training segment correl. w/ordinary		
TRAINA			% agreement - training w/ordinary		
TRAIND			% disagreement - training w/ordinary		

COMMON STORAGE ALLOCATION

Name INDX Size 1003 Page 1 of 1

Function Index to scratch DA file, TACQ, plus index to index

Name	Dimension	Format	Description	Symbol	Units
INDEX	2000		Index to TACQ (needed for CDC version)		
IPOINT	2001		Index to IPNT2		
IPNT2	2001		Index to INDEX		
IPEND			Pointer to last word of IPOINT filled		
IPIN			Pointer to last record read, in IPOINT		
			Indexing works		
			given ISEG = segment ID		
			binary search of IPOINT to find N where		
			IPOINT(N) = ISEG		
			then IPNT2(N) = IN		
			where IN = record no. of record on TACQ		

COMMON STORAGE ALLOCATION

Name SEGTRU Size 16 Page 1 of 1

Function Holds 1 record from Segment Truth File, input file (SEGTRU)

Name	Dimension	Format	Description	Symbol	Units
COUN4			Country ID		
IREG4			Region ID		
IZONE4			Zone ID		
ISTR4			Strata ID		
ISUB4			Substrata ID		
ISEG4			Segment ID		
IT			0 = ordinary segment, 1 = training segment		
IPRIOR	6		Priority list = segment nos.		
ISPW			0 = winter, 1 = spring		
PT	2		True proportion of wheat		

COMMON STORAGE ALLOCATION

Name SIGEX Size 59 Page 1 of 1

Function Holds 1 record from input file, Signature Extension

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
COUN5			Country ID		
IREG5			Region ID		
IZONE5			Zone ID		
ZB	3, 2		Bias error, dim. 1 = type (wheat, mixed, other) dim. 2 = B1 or B2		
ZSIG	3, 2, 6		Std. dev. of error, dim 1 = type 2 = σ1 or σ2 3 = training ~~segment~~ priorities		

COMMON STORAGE ALLOCATION

Name TRAINS Size 1032 Page 1 of 1

Function Holds training segments - 1 actual record of scratch DA file TACQ (same as ACQUIS file record + extra information) + information for CAMSF record

Name	Dimen-sion	For-mat	Description		Sym-bol	Units
COUN7			Country ID	from ACQUIS record		
IREG7			Region ID			
IZONE7			Zone ID			
ISTRA7			Strata ID			
ISUB7			Substrata ID			
ISEG7			Segment ID			
ITWIN	4, 25		Up to 25 acquisition dates for 4 windows			
ITTOT			Total no. acquisition dates			
TMM	3, 4, 25		M values (multi-temporal error)	saved from CAMS calculations		
TBB	3, 4, 25		Bias values			
TVV	3, 4, 25		Variance values			
TPTRUE			True proportion wheat	for (CAMSF) output file		
TIZULU	4		Acquisition dates			
TPEST	4		Estimated proportions of wheat			
TPERR	4		Error in estimates			
TERTOT	3		Error total - calculated from TM, TB, TV	needed for calculations - not part of TACQ record		
TM	3					
TV TB	3 3					

File Description

CAMS DA SCRATCH FILE - TACQ

This file is generated in CAMS to store training segment information to use .ater in CAMS.

Access Method: Direct with fixed length records - uses FORTRAN V direct access routines.

Status: Temporary, regenerated everytime CAMS run.

Sort: By I, I=1, 2000 for up to 2000 training segments. These are indexed by segment ID in array IPOINT, COMMON /INDX/, then array IPNT2 to get the actual index.

Media: Disk - FASTRAND

Record Formats: No header or trailer.

Record Length: 1020 words

Blocking Factor: 1

File Size: 2, 040, 000 words, assuming a maximum of 2000 training segments.

Detail of 1 Record:

- Country ID - 4 bytes, 4 alpha characters
- Region ID - 1 word integer, 3 digit no., 1 to 10
- Zone ID - 1 word integer, 3 digit no., 1 to 100
- Strata ID - 1 word integer, 4 digit no., 1 to 500
- Substrata ID - 1 word integer, 4 digit no., 1 to 3200
- Segment ID - 1 word integer, 5 digit no., 1 to 4000
- For each of 4 Crop Windows:
  - 25 entries for
    - Acquisition Date - 1 word integer, Zulu date
- Total No. of Accesses - 1 word integer, 3 digit no.

Multi-temporal error factors:

for wheat - 1 word, floating point, 0-1

for mixed - 1 word, floating point, 0-1

for other - 1 word, floating point, 0-1

Bias error factors:

for wheat - 1 word, floating point

for mixed - 1 word, floating point

for other - 1 word, floating point

Variance error factors:

for wheat - 1 word, floating point

for mixed - 1 word, floating point

for other - 1 word, floating point

True proportion of wheat this segment, flt. pt., % 0-100

Zulu Acquisition Day - 1 word integer (zero for no acquisition)

Estimated Proportion of Wheat - Flt. pt.

Error in Proportion of Wheat Estimate, flt. pt.

One ordered set for each of 4 windows

YES COMMON BLOCKS

COMMON STORAGE ALLOCATION

Name YESOT Size ______ Page 1 of 1

Function One record of YES output file

Name	Dimension	Format	Description	Symbol	Units
CID		R	4 character country ID		
IREGID		I	Region ID 1-10		
IZONID		I	Zone ID 1-100		
ISTRID		I	Strata ID 1-500		
YSTR		Real	True yield, range 0-99.99		Quintals / Hectare
IZPRDD	6	I	Zulu prediction date		
YSCI	6	Real	Estimated yield, range 0-99.99		Quintals / Hectare
VSYCI	6	Real	Standard deviation of yield error, range 0-99.99		Quintals / Hectare

COMMON STORAGE ALLOCATION

Name YESIN Size ______ Page 1 of 1

Function Store record from YES input file

Name	Dimen-sion	For-mat	Description	Sym-bol	Units
COUN		R	4 character country ID		
IREG		I	Region ID 1-10		
IZONE		I	Zone ID 1-100		
ISTRAT		I	Strata ID 1-500		
YTRUE		R	True yield		Quintals/Hectare
IZULU	6	R	Zulu prediction date - 6 points		
BIAS	6	R	Bias error - 6 prediction points		Quintals/Hectare
SD	6	R	Standard deviation - 6 prediction points		Quintals/Hectare

LOCAL STORAGE ALLOCATION

Name YES Size Page 1 of 1

Function

Name	Dimension	Format	Description	Symbol	Units
YNAME	2	R	3HYES for header record		
IFILL		I	0, for 0 fill of header record		
ZZZZ		R	4HZZZZ for trailer record		

PART III

LIST OF SUBROUTINES AND SUBROUTINE CALL STRUCTURE

LEM Subroutine Call Structure

- LEM
  - START
  - INPUT
    - ERRMES
    - INPCHK
      - ERRMES
    - EJECT
    - CAMSIN
      - EJECT
      - ERRMES
      - PAGER
    - CASIN
      - EJECT
      - ERRMES
        - PAGER
        - INPERR
        - CAMERS
        - CASER1
        - CAMER2
        - CASER2
        - SIGERR
        - WRAPUP
          - RANACF
          - EJECT
      - LFPA
      - RANACF
    - PAGER
  - INIT
  - ERRMC
  - SETPRF

- STG
  - ERRMES
  - BETAD
    - RDMIA
    - IBETAI
      - ALGAMA
- CAMS
  - EJECT
  - INITI
    - ERRMES
    - TSAVE
      - RANACF
      - ERRMES
      - SORTAG
  - INPT
    - ERRMES
    - TSAVE
  - REPORT
    - PAGER
    - FZULU
  - MULTI
  - CROP
    - BETAD
  - CLASS
    - BETAD
  - TSAVE
  - CORREL
    - TSAVE
  - SGEXT
    - BETAD
    - ERRMES
- YES
  - ERRMES
  - EJECT
  - FZULU
  - PAGER
  - BETAD

- CAS
  - CASPP
    - CLASSN
      - SEGTAB
      - DETCLS
      - ASSCLS
    - CASINL
    - GETYS
      - ERRMES
    - ERRMES
    - DS123
      - ERRMES
      - GROUP
        - ERRMES
      - BETAD
    - DS456
      - RANACF
    - DS7
      - RANACF
    - RANACF
    - CAS2
      - PAGER
      - RANACF
      - DS10
        - RANACF
        - RDM1A
        - ERRMES
        - RWCASF
          - RANACF
        - CASOUT
          - APHDR
            - EJECT
          - PAGER
      - CONFL
      - RWCASF
      - RWDISF
        - ERRMES
        - RANACF
      - CASOUT
      - DS18
        - YSUB
    - CAS3

CASPP (cont'd)

SUMREP

EJECT

PAGER

WRAPUP

RANACF

EJECT

PART IV

SUBROUTINE DESCRIPTION AND FLOWCHARTS

List of Subroutines in LEM

	Name	Function
1.	LEM	Main driver for LACIE error model.
2.	EJECT	Restores page and prints the page header.
3.	ERRMC	Initializes the random number seeds for each error source.
4.	ERRMES	Controls the printing of all error messages for LEM.
5.	FZULU	To convert Zulu date to year, month and day.
6.	PAGER	Automatic paging control routine.
7.	RANACF	Standardized random access I/O routine.
8.	SETPRF	Controls the printing of reports by subprogram and module.
9.	CASIN	This routine reads in and checks the CAS control cards.
10.	CASER1	This routine contains the input error messages for CAS.
11.	INIT	Initializes random number seeds.
12.	INPCHK	Checks the validity of the input parameters on the LEM control cards and checks the header records of all input files for valid case numbers.
13.	INPERR	Prints error messages for the LEM input processor.
14.	INPUT	Reads and checks LEM control card input and controls reading of all other data cards.
15.	LFPA	Given month, day and year, this routine returns the Zulu date.
16.	START	Initializes storage, flags and counters.
17.	STG	Segment truth generator subprogram for LEM.
18.	WRAPUP	This routine writes header records on CAS output files and prints status information at end of run.
19.	STGERR	Prints error messages for the segment truth generator.
20.	CAMSIN	Reads and checks CAMS control cards.

	Name	Function
21.	CAMSERS	Prints out CAMS control card error messages.
22.	BETAD	Controls the calculation of the incomplete beta function.
23.	IBETAI	Computes the incomplete beta function integral.
24.	ALGAMA	Computes the gamma function.
25.	RDM1A	Uniform random number generator CAMS subprogram subroutine set.
26.	CAMS	Driver for the CAMS subprogram which calculates the estimated proportion of wheat.
27.	REPORT	Prints the CAMS report.
28.	INITI	This routine initializes the input files and output files.
29.	CORREL	This routine tries to correlate a training segment with the ordinary segment being processed.
30.	MULTI	This routine calculates the multi-temporal error for training segments.
31.	SGEXT	This subroutine calculates the signature extension error for ordinary segments.
32.	CROP	This subroutine calculates the crop calendar error for training segments.
33.	TSAVE	This subroutine handles the I/O for the scratch RA file TACQ for CAMS.
34.	CLASS	This subroutine calculates the input classification error for training segments and the total classification error.
35.	INPT	This subroutine gets the next set of records to process from the input files.
36.	CAMER2	This subroutine contains the processing error messages for the CAMS module.
37.	YES	This subroutine calculates the estimated yield from the true yield.
38.	CAS	Main driver for the CAS simulator.
39.	APHDR	This routine prints the headers for the area and production summary report.

	Name	Function
40.	CASER2	This routine prints the processing error messages for CAS simulators.
41.	CASINL	This routine performs initialization tasks for each prediction point.
42.	CASINT	This routine performs miscellaneous tasks for the CAS simulator.
43.	CASPP	This routine performs the first pass CAS computations generating data sets 1-9.
44.	CASOUT	This routine prints the area and production report and saves data for the country report.
45.	CAS2	This routine generates data sets 10-17, 19, using data sets 1-9 read from the CAS intermediate file.
46.	CONFL	This routine computes the confidence levels in data set 13.
47.	DS123	This routine processes data sets 1, 2 and 3 at the sub-strata level.
48.	DS456	This routine processes data sets 4, 5 and 6 at the strata level.
49.	DS7	This routine processes data set 7 at the zone level.
50.	DS10	This routine processes data set 10 at the strata level.
51.	DS18	This routine computes CLWA and CLPRD in data set 18 on the final iteration.
52.	GETYS	This routine reads strata data from YESOUT file and obtains the proper value of estimated yield for the current bio-window or prediction date.
53.	GROUP	This routine reads segment data from the CAMS output file, selects the estimated proportion wheat for the proper bio-window for each segment, and aggregates the segment data up to the substrata level.
54.	PSUB	This function computes function P(X) for confidence level calculations.
55.	RWCASF	This routine reads a data set from the CAS cumulative file or writes a data set onto the CAS cumulative file.
56.	RWDISF	This routine reads and writes data from/onto the CAS distribution file.

	Name	Function
57.	SUMREP	This routine prints the CAS country summary report.
58.	TSUB	This routine computes the quantity T from Equation 39.
59.	YSUB	This function computes the quantity Y used in the confidence level calculations.
60.	CLASSN	This routine controls the computation of class numbers for all zones.
61.	SEGTAB	This routine forms the segment tables to be used to determine class.
62.	DETCLS	This routine determines how many classes and how many data points in each class.
63.	ASSCLS	This routine assigns the class number to each substrata in a zone.
64.	CAS3	This routine generates data sets 10-19 on final pass.

SUPPLIED UTILITY ROUTINES

Routine Day

Call Day (IYMD, IDAY)

Given IYMD (3) where IYMD (1) IS Day No.

IYMD (2) IS Month No.

IYMD (3) IS Year No.

Compute year day no. in IDAY

Routine PIMOD

Call PIMOD (A)

Convert ±A in radians to an angle $0-2\pi$

Routine SOL (Entry ALPHA)

Call ALPHA (IFLAG)

For emphemeris usage as called by hector

computes ALPHAM and ALPHAT and IFLAG = 1

Routine PAGER (Entry Eject)

Call PAGER (NLINES)

Updates line count in NLINE with NLINES

NPAGE = 0 causes page to be restored prior to print.

NPAGE - page no.

HEADER- 80 char. 20A5

ICASE- case no.

KO - 6 print unit

INMAX is max no. of lines allowed

Initially NLINE should be set > LINMAX and NPAGE = 0

SUPPLIED UTILITY ROUTINES
(CONTINUED)

Call EJECT (NLINES)

Causes page to be restored automatically and then prints headers.

Routine CLDAY

Call CLDAY

Given IDAY-DAY no. of the year compute in LMO-the month and in LDA the day no.

Need: IYEAR = 0 - Leap Year, ≠ 0 not Leap Year

Routine KEPLER

Call KEPLER (XM, XECC, XE, ERROR)

Given XM - Mean anomaly, XECC - eccentricity

Compute: E-eccentric anomaly, error = 0 means OK

Routine LFPA

Call LFPA [FLDA, LMO, LYR, ALFGM (can be dummy), DAYS]

Given: FLDA - day of month no., LMO - month no.,

LYR - year no. compute ALFGM - right ascension and

DAYS - Zulu day no.

Routine DEGMOD

Call DEGMOND (RAD, IDEG)

Given: angle rad in radians store the angle in deg., min., sec., in IDEG(1) - (3).

Routine FZULU

Call FZULU (IOATE, IOUT)

Given Zulu date in IDATE, compute year, month and day in IOUT(1) - IOUT(3).

Routine RDMIA

Call RDMIA(FL, U)

Given double precision random no. seed in FL, comoute random no. U (0-1) based on uniform distribution.

SUBROUTINE LEM

Purpose:

The subroutine LEM is the main driver for the LEM program. It defines all global common blocks used in the LEM program and calls the drivers for the various subprograms within LEM (INPUT, STG, CAMS, YES, and CAS).

Input:

Quantity	Common Block	Source
NFATAL	ARGLST	INPUT STG CAMS YES CAS
NTRIAL	LEMCM	INPUT
RSTART	LEMCM	INPUT
IPRINT	LEMCM	INPUT
ISTG	LEMCM	INPUT
ICAMS	LEMCM	INPUT
IYES	LEMCM	INPUT
IPRCAM	LEMCM	INPUT
IPRYES	LEMCM	INPUT
IPRCAS	LEMCM	INPUT

Output:

Quantity	Common Block	Destination
NSTART	CNTRL	CAS and related routines
ITER[illegible] NT	STATS	STG CAMS YES CAS and related routines

Linkage:

LEM, being the main program is called by the operating system.

Subroutines Used:

START	SETPRF
INPUT	STG
ERRMES	CAMS
INIT	YES
ERRMC	CAS
	WRAPUP

Processing:

LEM calls START to initialize a few flags and counters, then calls INPUT to read and check all control card input data. If any fatal input errors are detected in the input data, subroutine ERRMES is called to abort the run.

If no fatal input errors are detected, then subroutine INIT is called to initialize the random number seeds.

Next for each Monte Carlo iteration (starting with iteration number RSTART + 1 and continuing through iteration NTRIAL) the following subroutines are executed in order:

ERRMC	sets random number seeds
STG	Segment Truth Generator
CAMS	CAMS Simulator
YES	Yield Estimation Model
CAS	CAS Simulator

(Prior to each of the calls to STG, CAMS, YES, and CAS, LEM calls subroutine SETPRF to properly set the print flag PRINTF.)

Finally after the last Monte Carlo iteration has been completed subroutine WRAPUP is called to print the program status information and to close random access files.

SUBROUTINE ERRMC

Purpose:

Subroutine ERRMC provides the Error Model control for the LEM program by properly initializing the random number seeds for the following error sources:

- Classification error
- Signature extension error
- Segment Crop Calendar error
- CAS Group II error "Most Recent Non-Epoch Year" Historical Proportion of Wheat
- CAS Group III Multi-year Proportion of Wheat

Input:

Quantity	Common Block	Source
ICAMS	LEMCM	INPUT
ICLASS	LEMCM	INPUT
ISEXT	LEMCM	INPUT
ISCC	LEMCM	INPUT
ICAS	LEMCM	INPUT
ICAS2	LEMCM	INPUT
ICAS3	LEMCM	INPUT
RSEED	LEMCM	INPUT

Output:

Quantity	Common Block	Used By
SEED	CNTRL	STG, CAMS, YES, CAS, WRAPUP

Linkage:

CALL ERRMC

There are no arguments. All input/output quantities are transmitted through COMMON storage.

Subroutine Used:

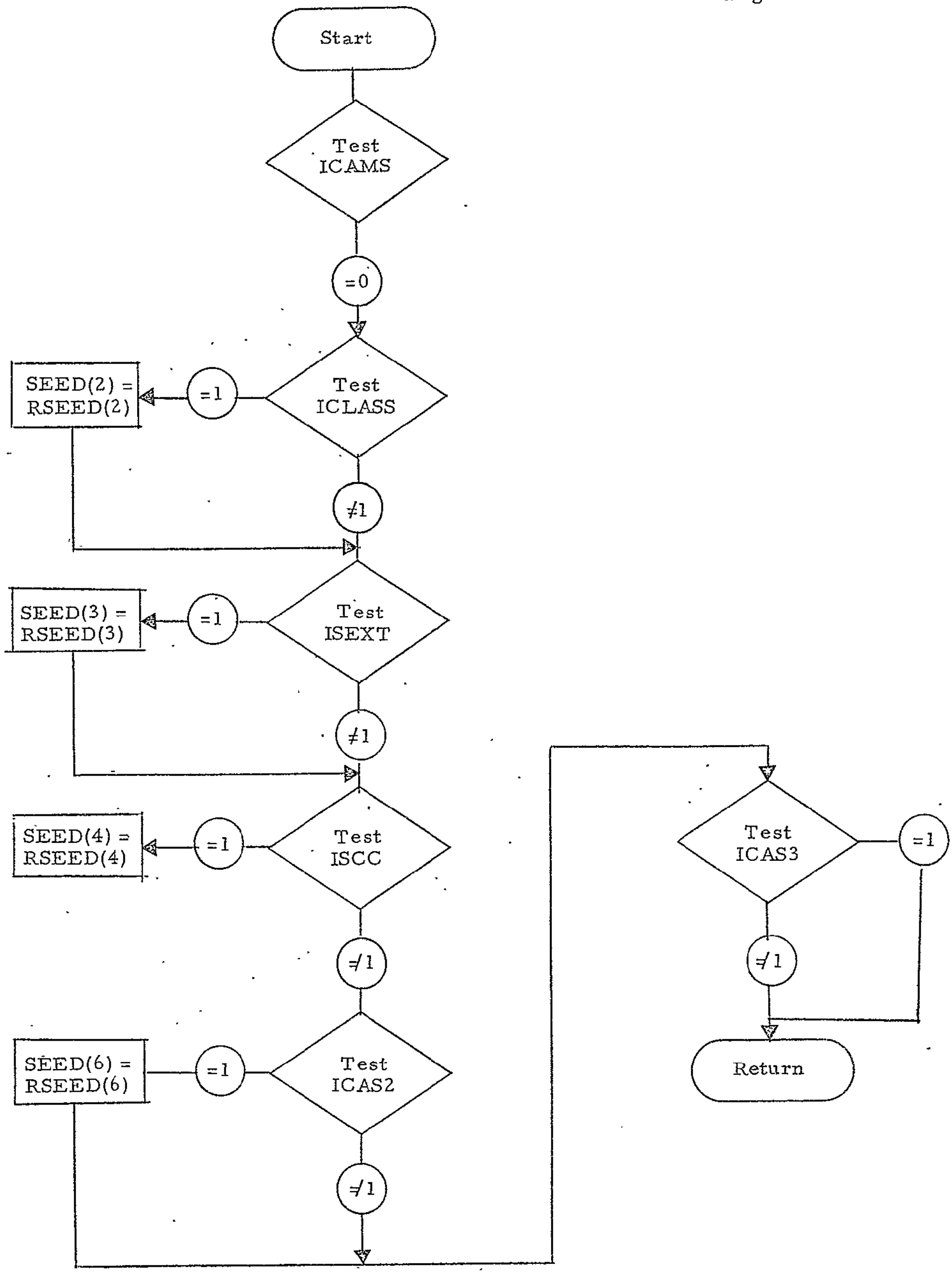
None.

Local Variables:

None.

Processing:

See flow chart.


Start
Test ICAMS
=0
Test ICLASS
=1
SEED(2) = RSEED(2)
≠1
Test ISEXT
=1
SEED(3) = RSEED(3)
≠1
Test ISCC
=1
SEED(4) = RSEED(4)
≠1
Test ICAS2
=1
SEED(6) = RSEED(6)
≠1
Test ICAS3
=1
≠1
Return

SUBROUTINE ERRMES

Purpose:

Subroutine ERRMES prints an error message as directed by the input parameters.

Input:

PROG, SUBR, ICODE, and LEVEL are input parameters transmitted through the calling sequence. In addition, the following quantities are passed through COMMON storage:

Quantity	Common Block
NERRS	ARGLST
NFATAL	ARGLST
NPERRS	ARGLST
NARG	ARGLST
ARG	ARGLST

Output:

Quantity	Common Block
NERRS	ARGLST
NFATAL	ARGLST
NPERRS	ARGLST

In addition to the error counters being advanced, an error message is written onto the printed report. Parameters obtained from the array ARG may be included in the error message.

Linkage:

CALL ERRMES(PROG, SUBR, ICODE, LEVEL)

where

PROG is the subprogram name in A6 format (e.g., 4H CAMS, 3H YES, etc.)

SUBR — is the name of the subroutine within the subprogram also in A6 format (e.g., INPCHK)

ICODE — is the error code. (See processing for a list of codes and associated messages.)

LEVEL — is the level of the error.
- = 0 for non fatal
- = 1 for fatal
- = 2 for step fatal - skip this step, e.g., CAMS, but continue with the run.

In addition, a list of arguments (to be printed out as part of the error message) is stored in the array ARG within the COMMON block ARGLST and the argument count is stored in NARG.

Subroutines Used:

PAGER	CAMER2
INPERR	CASER2
CAMERS	STGERR
CASER1	WRAPUP

Local Variables:

IMES — Error code

BLANK — Word of blanks (format 1H)

NONFTL — Part of error message (3 H NON)

ERRLVL — Blank or = NONFTL (used to fill part of error message)

Processing:

For non-fatal errors the following general message is printed out on the report followed by a specific error message:

***** NONFATAL ERROR IN SUBPROGRAM ______ SUBROUTINE ______ ERROR CODE __ *****

For step fatal errors the following general message is printed out on the report followed by a specific error message:

***** FATAL ERROR n IN SUBPROGRAM ______ SUBROUTINE ______
ERROR CODE __ *****

Subroutine RANACF

Purpose:

RANACF is a standardized random access I/O subroutine. Written in Fortran, it provides a central location for all operations on random access files (opening, closing, reading, and writing). RANACF contains calls to the installation-dependent random access routines.

Input:

IFILE, IREC, N, L, IOPT (See Linkage)

In addition, if IOPT = 2, then BUF is input to RANACF.

Output:

If IOPT = 1, then BUF is output by RANACF.

Linkage:

CALL RANACF (IFILE, IREC, BUF, N, IX, L, IOPT)

where

IFILE	=	Logical unit number of the random access file.
IREC	=	Record number to read or write.
BUF	=	Array of N words to be read from or written onto the random access file.
N	=	Number of words to read or write.
IX	=	Index array (length L). (Required on CDC computer but not on UNIVAC.)
L	=	Length of index.
IOPT	=	Entry point option:
	=0	to open the file
	=1	to read a record
	=2	to write a record
	=-1	to close the file

Subroutines Used:

OPENMS READMS WRITMS CLOSEMS	Used on CDC computer. Similar routines are required on UNIVAC.

Local Variables:

None.

Processing:

The appropriate routine is called to open, close read, or write the file as specified by IOPT.

SUBROUTINE SETPRF

Purpose:

Subroutine SETPRF sets the print flag PRINTF to print reports or suppress printing for a given module depending upon the iteration number and the input print flag for that module.

Input:

Quantity	Common Block	Source
IPR	(see Linkage)	LEM
RSTART	LEMCM	INPUT
NTRIAL	LEMCM	INPUT
ITER	STATS	LEM

Output:

Quantity	Common Block	
PRINTF	CNTRL	{ = 0 to suppress printing = 1 to print this iteration

Linkage:

CALL SETPRF (IPR)

where

IPR is the input print flag (e.g., IPRINT, IPRCAM, etc.)

= 0	to print first and last iterations of each run
= 1	to print every iteration
= 2	to print only the last iteration
= 3	to suppress all printing of reports

Subroutines Used:

None.

Local Variables:

None.

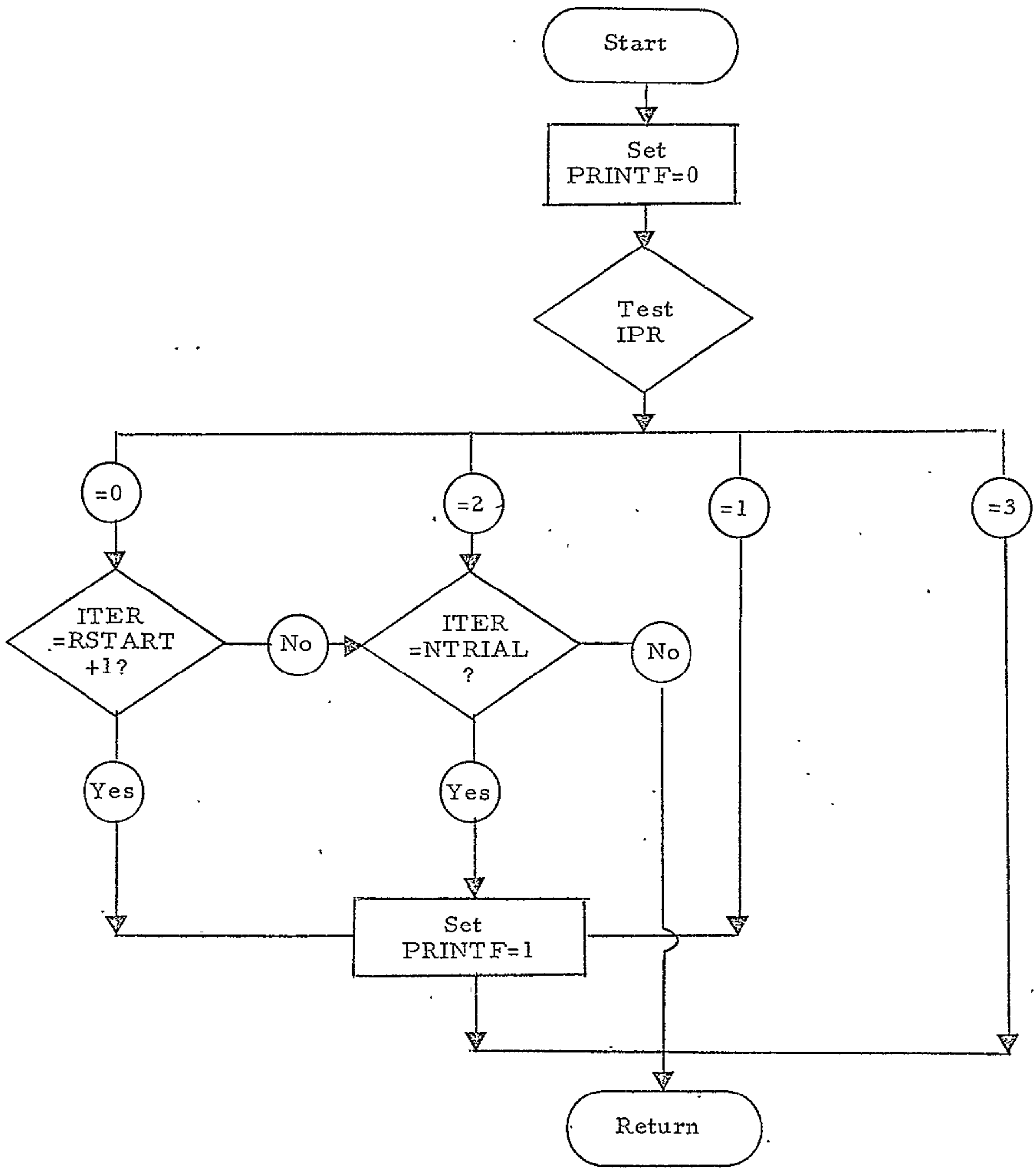
Processing:

See flow chart.

Subroutines Used:
None

Local Variables:
None

Processing:



SUBROUTINE INPCHK

Purpose:

Subroutine INPCHK checks the validity of the LEM Control Card parameters. It also reads the header records of each required input file and checks the file name, case number and country on that file.

Input:

All of the quantities in the COMMON blocks LEMCM and FILES are inputs to INPCHK.

In addition, the following quantities are input to INPCHK:

Quantity	Common Block	Source
NTRMX	CONST	Block Data
MAXR	CONST	Block Data
MAXZ	CONST	Block Data

Also, the following files may be input to INPCHK so the header information may be checked:

- Segment ID file
- Crop Window file
- Substrata Historical file
- CAMS Error Model file
- YES Error Model file
- Signature Extension file
- Data Acquisition file
- Segment Truth file
- CAMS Output file
- YES Output file
- CAS Cumulative Output file

Output:

Quantity	Common Block	Used By
NERRS	ARGLST	LEM, WRAPUP
NFATAL	ARGLST	WRAPUP

Linkage:

CALL INPCHK

There are no parameters in the calling sequence.

Subroutines Used:

ERRMES
RANACF

Local Variables:

FILL	Filler for header records of input files
NFILL	Number of words of filler necessary to complete record

Processing:

Check Input Data
(Control Card parameters and File Headers)

1. NTRIAL - RSTART $\leq$ NTRMX?
2. RSTART $<$ NTRIAL?
3. 0 $\leq$ STARTR $\leq$ ENDR $\leq$ MAXR?
4. 0 $\leq$ STARTZ $\leq$ ENDZ $\leq$ MAXZ?
5. ISTG, ICAMS, IYES, must be 0, 1, 2, or 3.
6. If CAMS $\neq$ 0, then ISTG $\neq$ 0?

7. If ISTG = 0 or if ISTG = 1 or 3 and RSTART = 0, then read and check Segment ID file header
   a) filename = "SEGMENT"
   b) case number = ICSESG
   c) country = CUNTRY
8. If ICAMS = 0 or if ICAMS = 1 or 3 and RSTART = 0, then read and check Crop Window file header.
   a) filename = "CROPWIND"
   b) case number = ICSECW
   c) country = CUNTRY
9. If ICAMS = 0 or if ICAMS = 1 or 3 and RSTART = 0, then read and check CAMS Error Model file header.
   a) filename = "CAMSERR"
   b) case number = ICSECE
   c) country = CUNTRY
10. If ICAMS = 0 or if ICAMS = 1 or 3 and RSTART = 0, then read and check Signature Extension file header.
   a) filename = "SIGEXTEN"
   b) case number = ICSESE
   c) country = CUNTRY
11. If ICAMS = 0 or if ICAMS = 1 or 3 and RSTART = 0, then read and check Data Acquisition file header.
   a) filename = "ACQUISI"
   b) case number = ICSEAC
   c) country = CUNTRY
12. If IYES = 0 or if IYES = 1 or 3 and RSTART = 0, then read and check YES Error Model file header.
   a) filename = "YESERROR"
   b) case number = ICSEYM
   c) country = CUNTRY

13. Read and check header of Substrata Historical file.
    a) filename = "SUBHIST"
    b) case number = ICSESH
    c) country = CUNTRY

14. If ISTG = 2 or if ISTG = 1 or 3 and RSTART > 0, then read and check header of Segment Truth file.
    a) filename = "SEGTRUTH"
    b) case number = ICSEST
    c) country = CUNTRY

15. If ICAMS = 2 or if ICAMS = 1 or 3 and RSTART > 0, then read and check header of CAMS Output file.
    a) filename = "CAMSOUT"
    b) case number = ICSECO
    c) country = CUNTRY

16. If IYES = 2 or if IYES = 1 or 3 and RSTART > 0, then read and check header of YES Output file.
    a) filename = "YES"
    b) case number = ICSEYS
    c) country = CUNTRY

17. Open CAS Cumulative Output file (a random access file).

18. If RSTART > 0, then read and check header of CAS Cumulative Output file.
    a) filename = "CASCUM"
    b) case number = ICASE
    c) country = CUNTRY
    d) NT = RSTART

Rewind each file before and after reading its header record.

SUBROUTINE INPUT

Purpose:

LEM reads the LEM control cards and calls INPCHK to check the LEM control card data. INPUT also calls CAMSIN and CASIN to read the remaining control card input data.

Input:

All of the quantities in Common block LEMCM are input to INPUT from the input file.

In addition the following quantities are inputs to INPUT.

Quantity	Common Block	Source
INP	FILES	Block Data
OUTP	FILES	Block Data
NERRS	ARGLST	ERRMES
NFATAL	ARGLST	ERRMES

Output:

All of the quantities in common block LEMCM.

In addition, the following quantities are output from INPUT.

Quantity	Common Block	Destination
ARG(1)	ARGLST	ERRMES
NPAGE	PAGECM	EJECT

Linkage:

CALL INPUT

There are no parameters in the calling sequence.

Subroutines Used:

ERRMES
EJECT
INPCHK
CAMSIN
CASIN
PAGER

Local Variables:

LBL	Label on LEM control cards
LBL1	Label on first LEM control card

Processing:

1. The first two LEM control cards are read in.
2. The labels on the first two LEM control cards are checked. They must be LEM 01 and LEM 02.
3. Next subroutine EJECT is called to eject a page on the output file and to write the case header at the top of the page.
4. The data from the first two LEM control cards is printed out.
5. Next the third and fourth LEM control cards are read, their labels are checked and the data is printed out.
6. Next subroutine INPCHK is called to check the LEM control card data for errors.
7. Then routines CAMSIN and CASIN are called to read and check the CAMS control card data and the CAS control card data.
8. Finally the number of non fatal and fatal errors detected in the LEM, CAMS, and CAS control cards is printed out.

SUBROUTINE STG

Purpose:

The purpose of the Segment Truth Generator (STG) is to generate the true proportion wheat and the true proportion mixed pixels for each sample segment. STG interfaces with the Segment ID file and the Substrata Historical file to obtain the data necessary to produce the Segment Truth file.

Input:

The following quantities are obtained from the Segment ID file:

COUN	Country ID
IREG	Region ID
IZONE	Zone ID
ISTRAT	Strata ID
ISUBS	Substrata ID
ISEG	Segment ID
ITRAIN	Training Segment Indicator
ITSPRL	Training Segment Priority List
SLAT	Segment latitude (not used)
SLONG	Segment longitude (not used)
GRIDNO	Grid Number (not used)
ISW	Sprin/Winter wheat indicator

The following quantities are obtained from the Substrata Historical file:

COUN2	Country ID
IREG2	Region ID
IZONE2	Zone ID
ISTRA2	Strata ID
ISUB2	Substrata ID
NSEG	Number of sample segments in this substratum
IDSEG	List of sample segments in this substrata (dimensioned IMXSEG)

GRPNO	Group number (not used)
HISTPW	Historical PW (not used)
AREA	Substrata land area (not used)
PWK	True proportion of wheat
NAGR	Number of agricultural segments in the substrata (not used)
NA	Number of allocated segments in the substrata (not used)
DELTPW	δPW = bias of true proportion of wheat (not used)
DELTPM	δPM = ratio of true mixed pixels
CV1	Coefficient of variation for year-to-year change in PW (not used)
CV2	Coefficient of variation for within county variation of PW
CV3	Coefficient of variation for within county variation of proportion of mixed pixels
CV4	Coefficient of variation of multi-year historical wheat area (not used)

The following input quantities are obtained from labeled COMMON:

Quantity	Common Block	Source
PRINTF	CNTRL	SETPRF
SEED(1)	CNTRL	ERRMC
ENDFIL	CONST	Block Data
IMXSEG	CONST	Block Data
ITSFLG	CONST	INPCHK
SEGID	FILES	INPCHK
LSEGID	FILES	INPCHK
SUBHST	FILES	Block Data
LSUBH	FILES	Block Data
SEGTRU	FILES	Block Data
LSEGTR	FILES	Block Data
ICASE	LEMCM	INPUT
CUNTRY	LEMCM	INPUT
NTRIAL	LEMCM	INPUT
RSTART	LEMCM	INPUT

Quantity	Common Block	Source
IPRINT	LEMCM	INPUT
STARTR	LEMCM	INPUT
STARTZ	LEMCM	INPUT
ENDR	LEMCM	INPUT
ENDZ	LEMCM	INPUT
ISTG	LEMCM	INPUT
ICASE1	LEMCM	INPUT
ICASE2	LEMCM	INPUT
ICASE3	LEMCM	INPUT
NLINE	PAGECM	PAGER
MXLINE		Block Data
ITER	STATS	LEM

Output:

The following quantities are written onto the Segment Truth file:

Variable	
ICASE	
ITSFLG	Training Segment flag
CUNTRY	Country ID
IREG	Region ID
IZONE	Zone ID
ISTRAT	Strata ID
ISUBS	Substrata ID
ISEG	Segment ID
ITRAIN	Training Segment Indicator
ITSPRL	Training Segment Priority List
ISW	Spring/Winter indicator
PWKI	PW_{ki} = True proportion wheat for this segment
PMKI	PM_{ki} = True proportion mixed pixels for this segment

In addition, the following quantities are printed on the Segment Truth report:

PWK	Substrata true PW
PWKI	
PMKI	
AVEPW	Average PW for current substrata
ERRPW	Error in segment PW (segment True PW - Substrata True PW)

The following output quantities are stored in COMMON:

Variable	Common Block	Used By
NREC(1)	STATS	WRAPUP
NSEGTR	STATS	WRAPUP

Linkage:

CALL STG

There are no arguments in the calling sequence. All input/output quantities are transmitted through COMMON storage.

Subroutines Used:

BETAD	Beta Distribution routine CALL BETAD (SEED, AVE, SIGMA, RN, IOPT, IERROR)
ERRMES	CALL ERRMES (PROG, SUBR, ICODE, LEVEL)
EJECT	Page Eject routine
PAGER	Automatic Paging routine

Processing:

A detail flow chart for the Segment Truth Generator is given on the following pages.

The true PW and true PM for each segment are computed as follows:

a) If ISTG = 3 (zero error case)

PWKI = PWK

PMKI = PWK * DELTPM

(i.e., segment truth values = substrata truth values)

b) If ISTG = 0 or 1

PWKI is computed by the BETAD subroutine with

mean = PWK/100.0

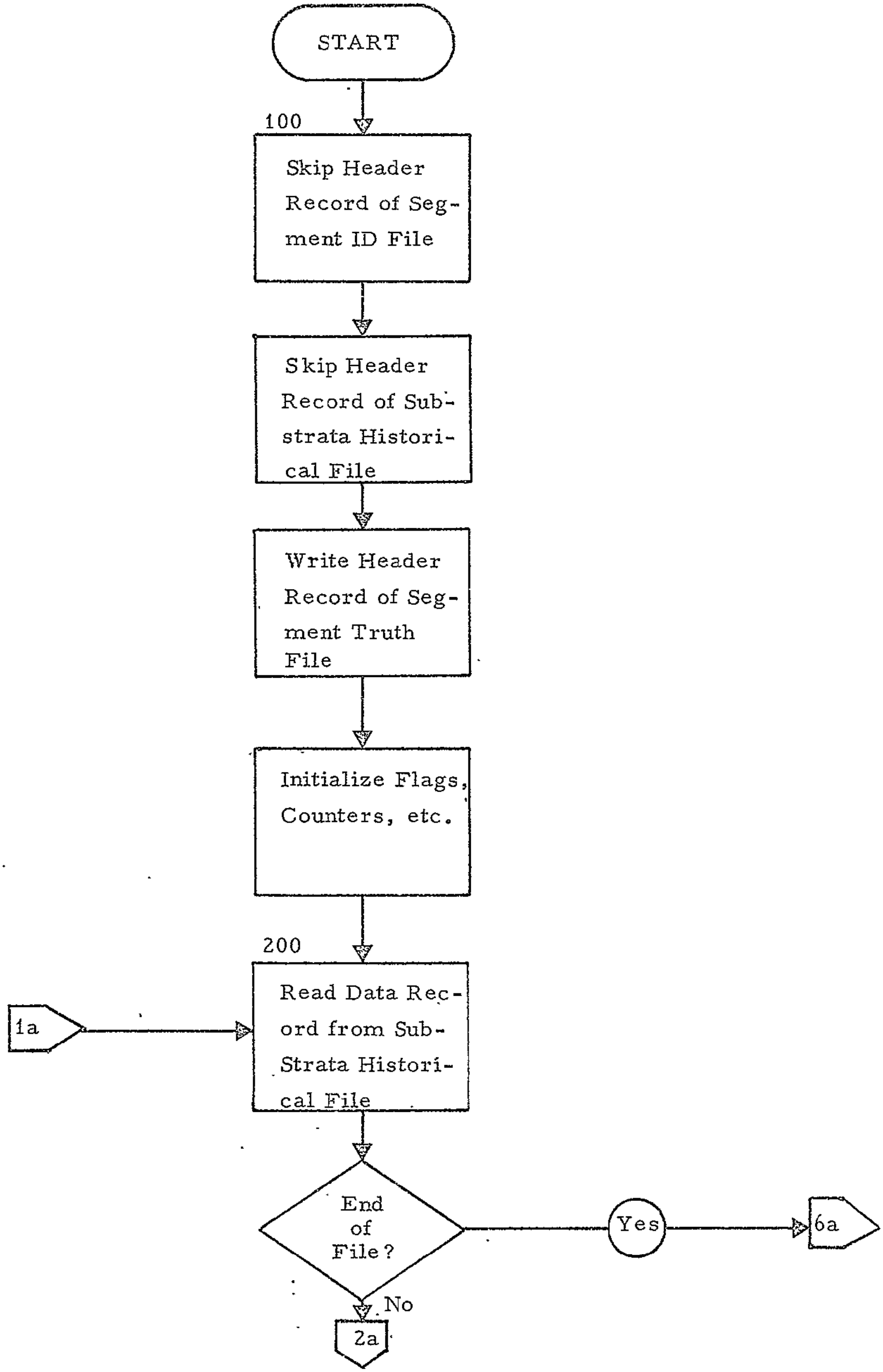
SIGMA = PWK * CV2/100.0

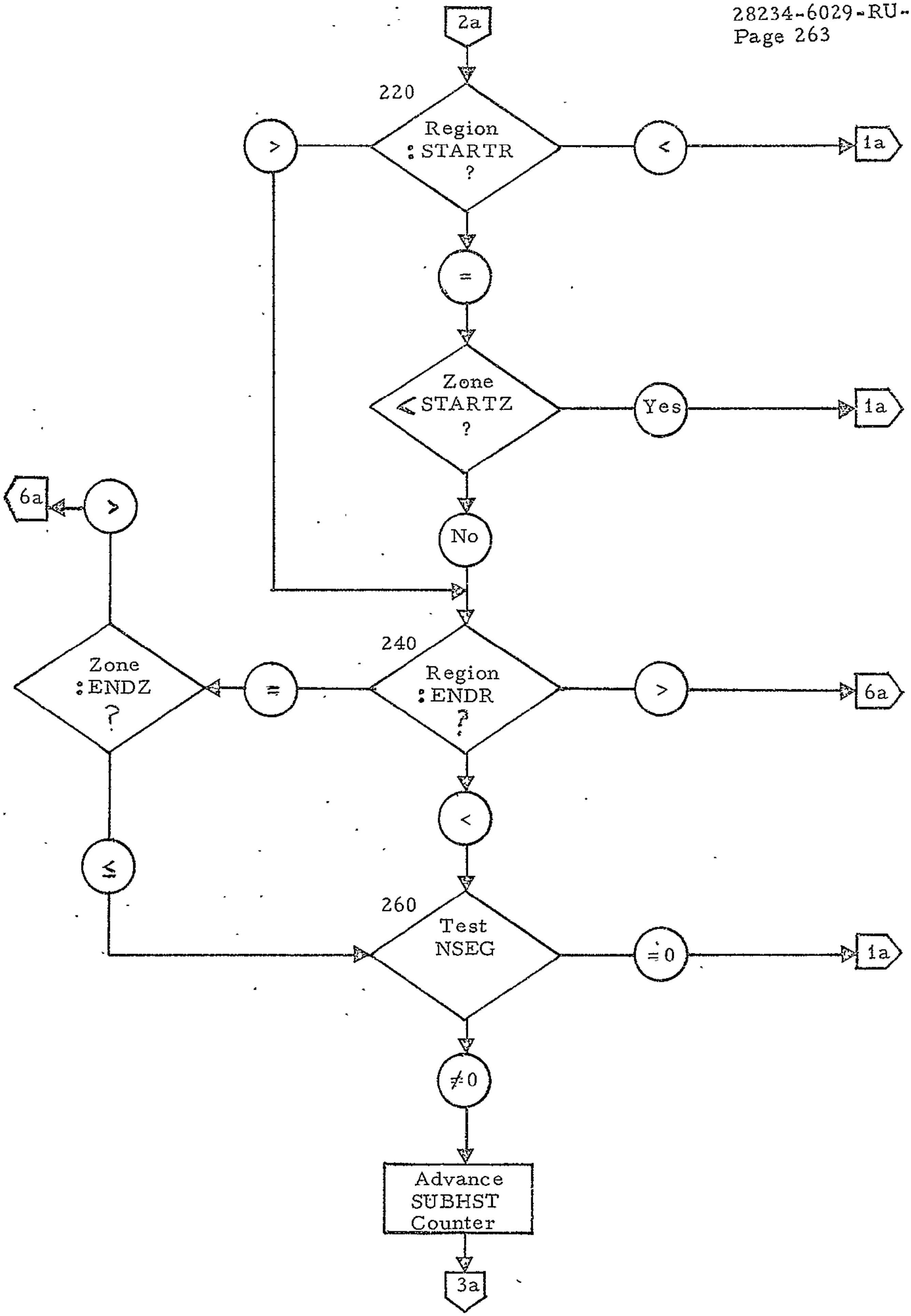
PMKI is computed by the BETAD subroutine with

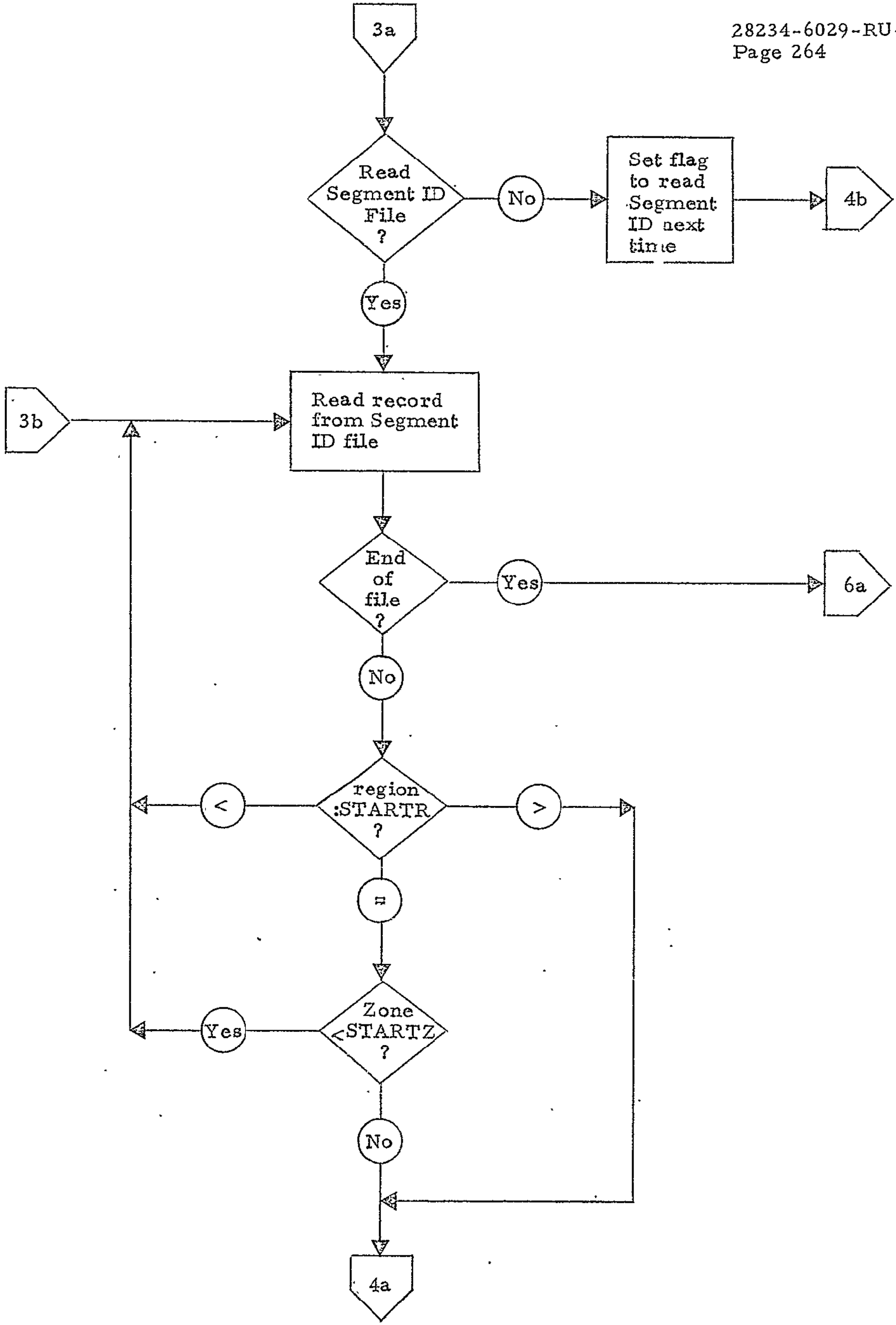
mean = (PWKI * DELTPM)/100.0

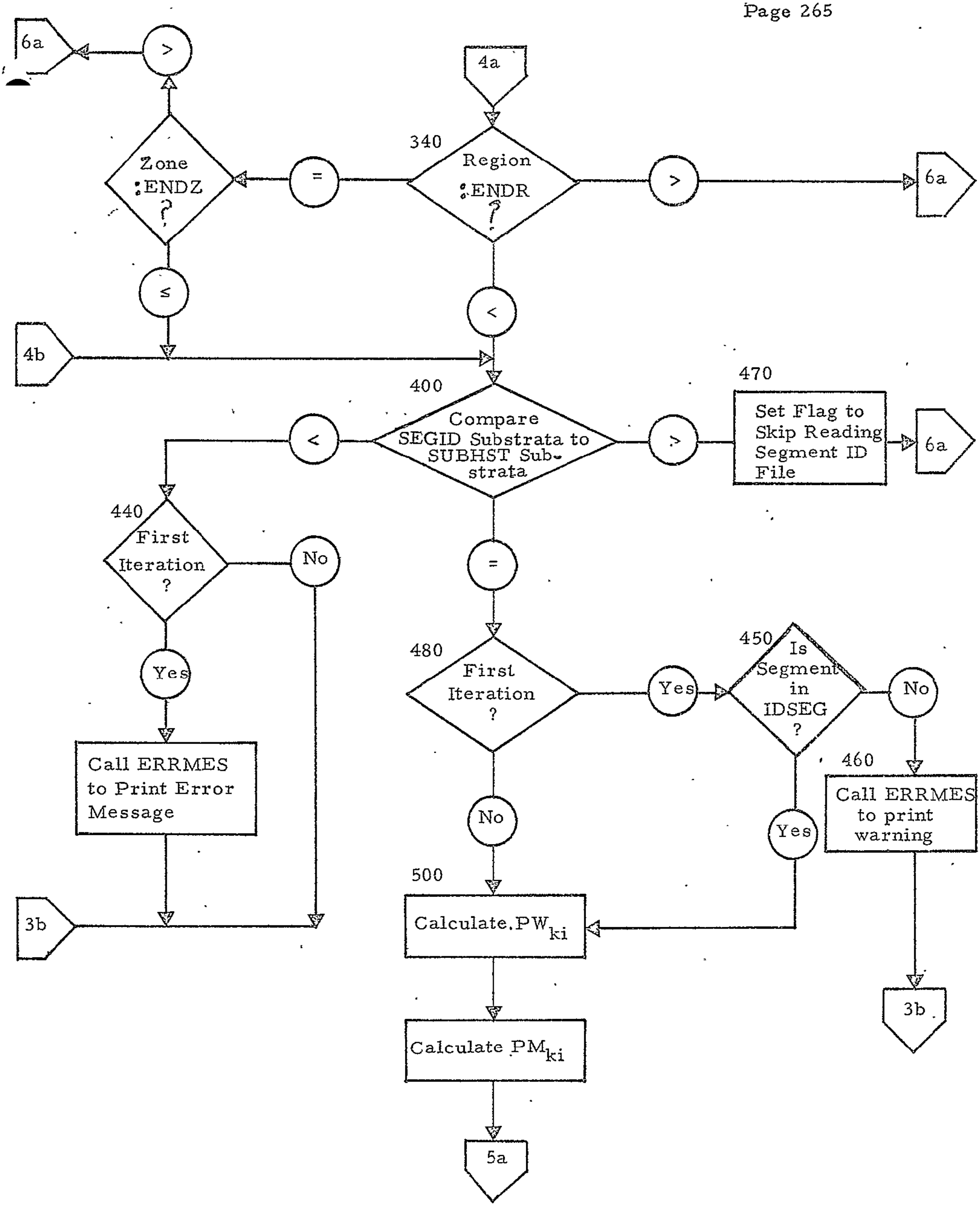
SIGMA = mean * CV3

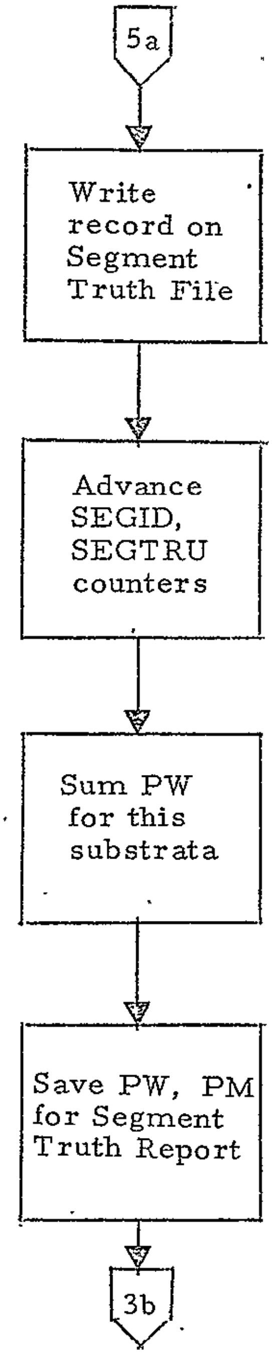
FLOW CHART OF SUBROUTINE STG

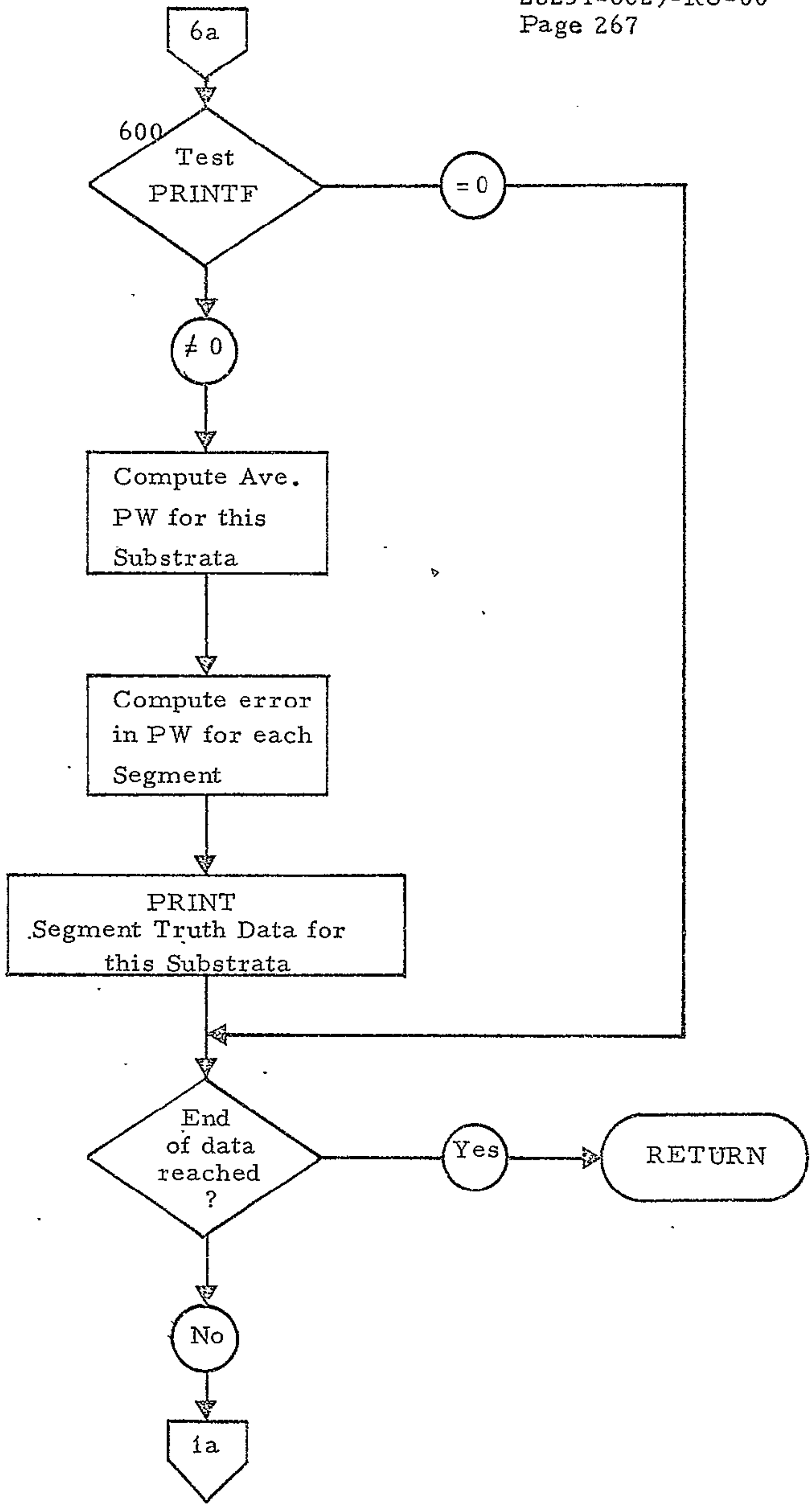



2a
220
Region
: STARTR
?
>
<
1a
=
Zone
≤ STARTZ
?
Yes
1a
No
6a
>
240
Region
: ENDR
?
=
>
6a
Zone
: ENDZ
?
<
≤
260
Test
NSEG
= 0
1a
≠ 0
Advance
SUBHST
Counter
3a


3a
Read Segment ID File ?
No
Set flag to read Segment ID next time
4b
Yes
3b
Read record from Segment ID file
End of file ?
Yes
6a
No
region :STARTR ?
<
>
=
Zone <STARTZ ?
Yes
No
4a


6a
>
4a
340
Region
:ENDR
?
Zone
:ENDZ
?
=
>
6a
≤
<
4b
400
Compare
SEGID Substrata to
SUBHST Sub-
strata
<
>
470
Set Flag to
Skip Reading
Segment ID
File
6a
440
First
Iteration
?
No
=
Yes
Call ERRMES
to Print Error
Message
480
First
Iteration
?
Yes
450
Is
Segment
in
IDSEG
?
No
460
Call ERRMES
to print
warning
No
Yes
500
Calculate PW$_{ki}$
3b
Calculate PM$_{ki}$
3b
5a


5a
Write record on Segment Truth File
Advance SEGID, SEGTRU counters
Sum PW for this substrata
Save PW, PM for Segment Truth Report
3b


6a
600
Test PRINTF
= 0
≠ 0
Compute Ave. PW for this Substrata
Compute error in PW for each Segment
PRINT Segment Truth Data for this Substrata
End of data reached ?
Yes
RETURN
No
1a

CAMS SUBROUTINE DESCRIPTION

SUBROUTINE CAMSIN

Purpose:

This subroutine reads in the CAMS control cards, echos the data on the printer, and stores it in the COMMON block /CAMSCM/. The data consists of 13 cards, one CAMS control card, eight multi-temporal sampling matrix cards, and four crop calendar coefficient cards. CAMSIN checks for errors in the data and, if found, prints appropriate messages and aborts.

Input:

The main inputs are the 13 input data cards. See the CAMS Problem Description, Figures 1-4, for the format and contents.

Also needed are the COMMON block quantities:

/FILES/	INP OUTP	input device number output device number
/PAGECM/		used by PAGER and EJECT subroutines
/LEMCM/	TITLE	used by PAGER and EJECT subroutines

Output:

The main output is the COMMON block /CAMSCM/.

/CAMSCM/	IMODEL IMULTI ISIGEX ISKIP ITMAX IREP IWIND	CAMS control card inputs; see Figure 1, CAMS Problem Description
	IGROUP(3, 2, 15)	multi-temporal sampling matrix data; see Figure 2, CAMS Problem Description - dimension 1 = type (wheat, mixed, other) 2 = season (winter, spring) 3 = for IGROUP, which M (1, 2, or 3)

MS(3, 2, 3)	for MS, values of M (M(2) and M(3) from input, M(1) ≡ 1) if model 1, ignore model 2 data if model 2, ignore model 1 data and store in type = wheat (type = mixed, other values set to 0)
G(3, 2, 2) H(3, 2, 2)	crop calendar coefficients data; see Figure 3, CAMS Problem Description - dimension 1 = type 2 = season 3 = for G, which G (G1 or G2) for H, which H (H1 or H2) if model 1, ignore model 2 data if model 2, ignore model 1 data and store in type = wheat only (mixed and other set to 0)

Also output to the printer is an echo of the input data, with the same quantities as are on the input data cards. Blank default columns will contain 0.0 values.

Also, possible output are the input error messages. See CAMS Problem Description, Section 5.2.

Linkage:

CALL CAMSIN

Subroutines Used:

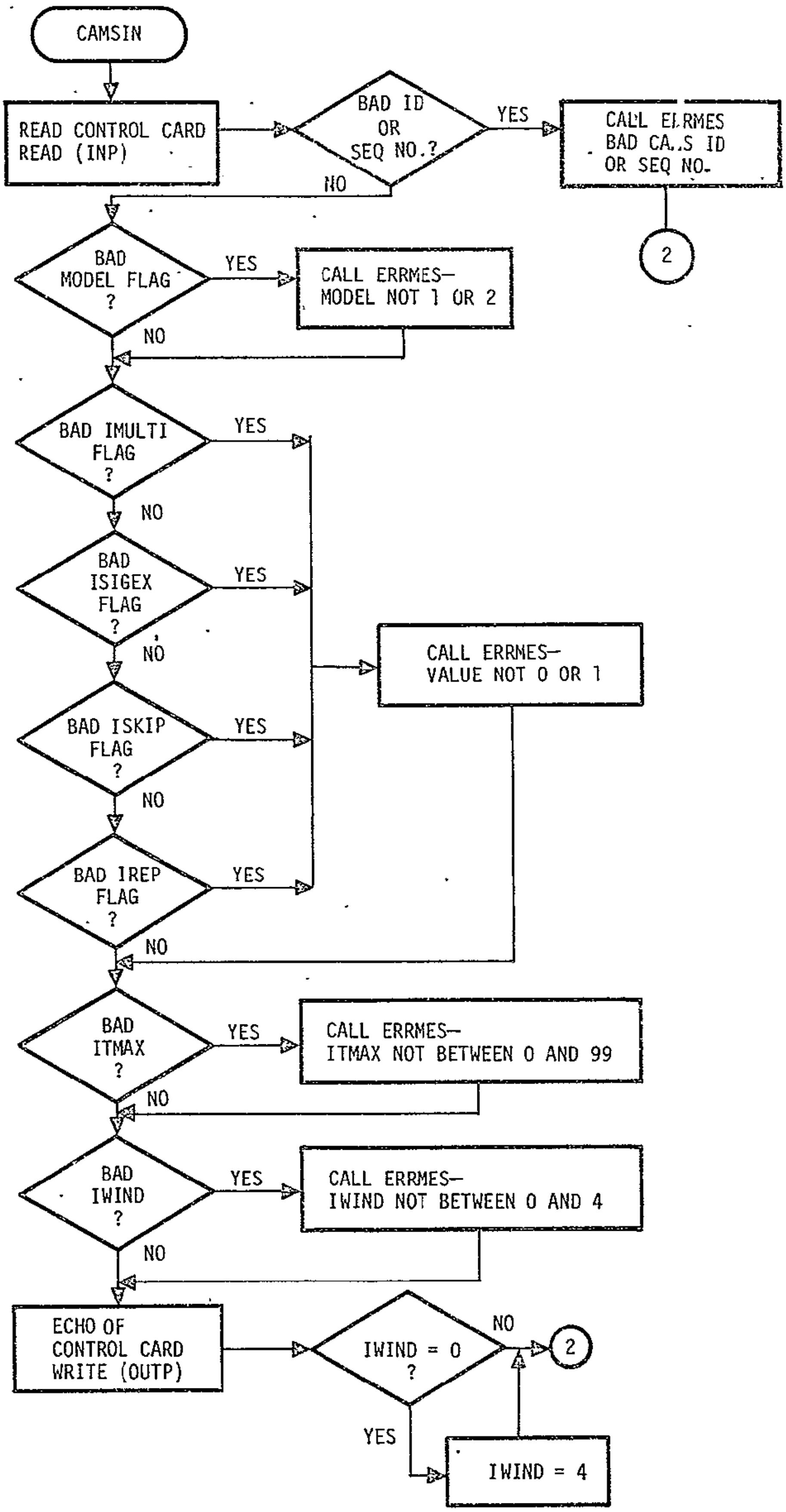
CALL EJECT(IND) CALL PAGER(IND)	to help printing echo of data - supplied utility routines
ABS(X)	absolute value

Local Variables:

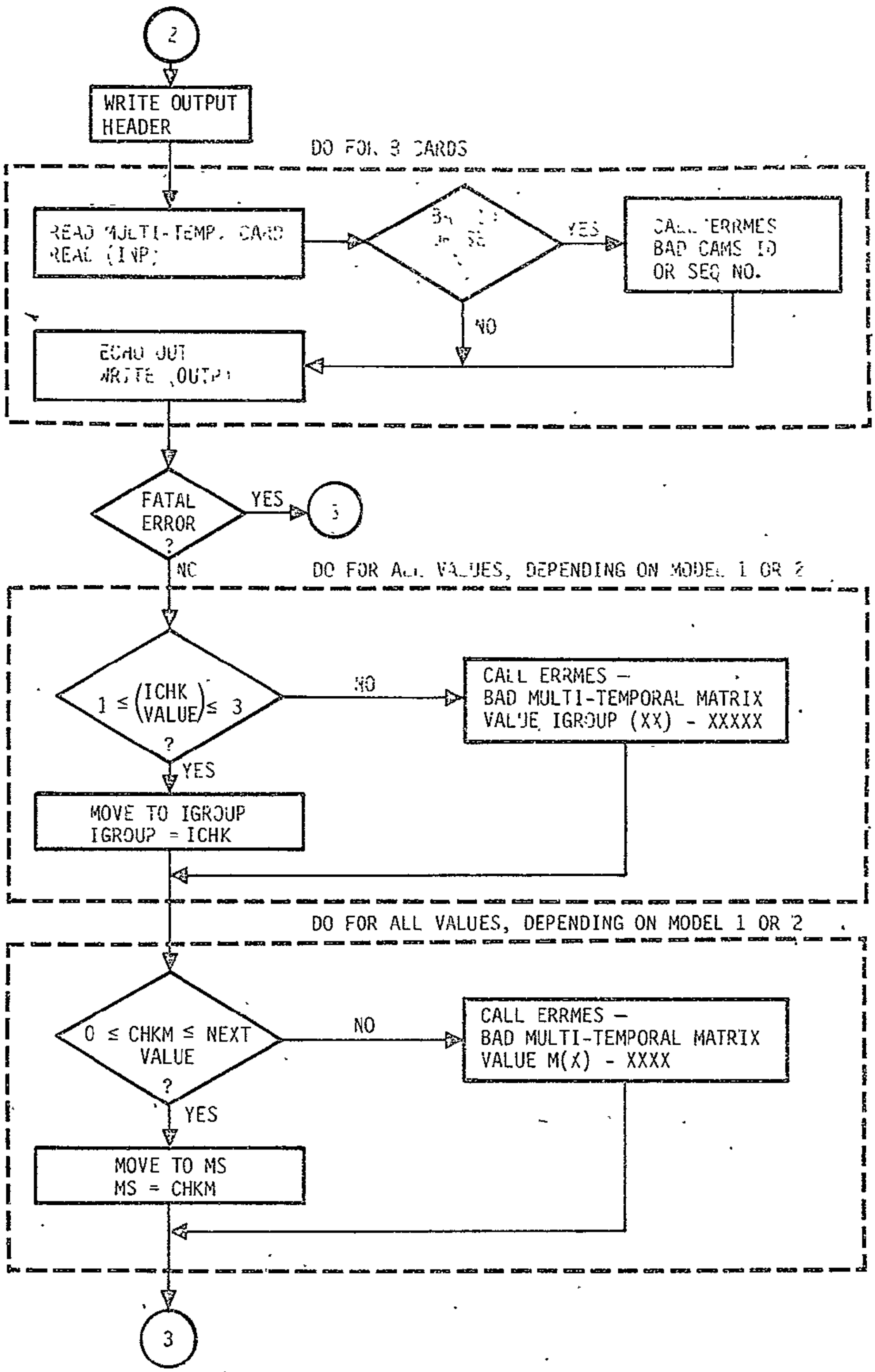
ICHK(4, 2, 15)	holds IGROUP array for error checking dimension 1 = type, model 1 wheat = 1 model 1 mixed = 2 model 1 other = 3 model 2 = 4 dimension 3 = which M to use, 1-4 ≡ 1 5-15 = input data
CHKM(4, 2, 3)	holds MS array for checking, M(1) ≡ 1
ISEQ(4, 2)	holds sequence numbers for matrix
CHKG(4, 2, 2) CHKH(4, 2, 2)	holds G and H arrays for error checking

C-4

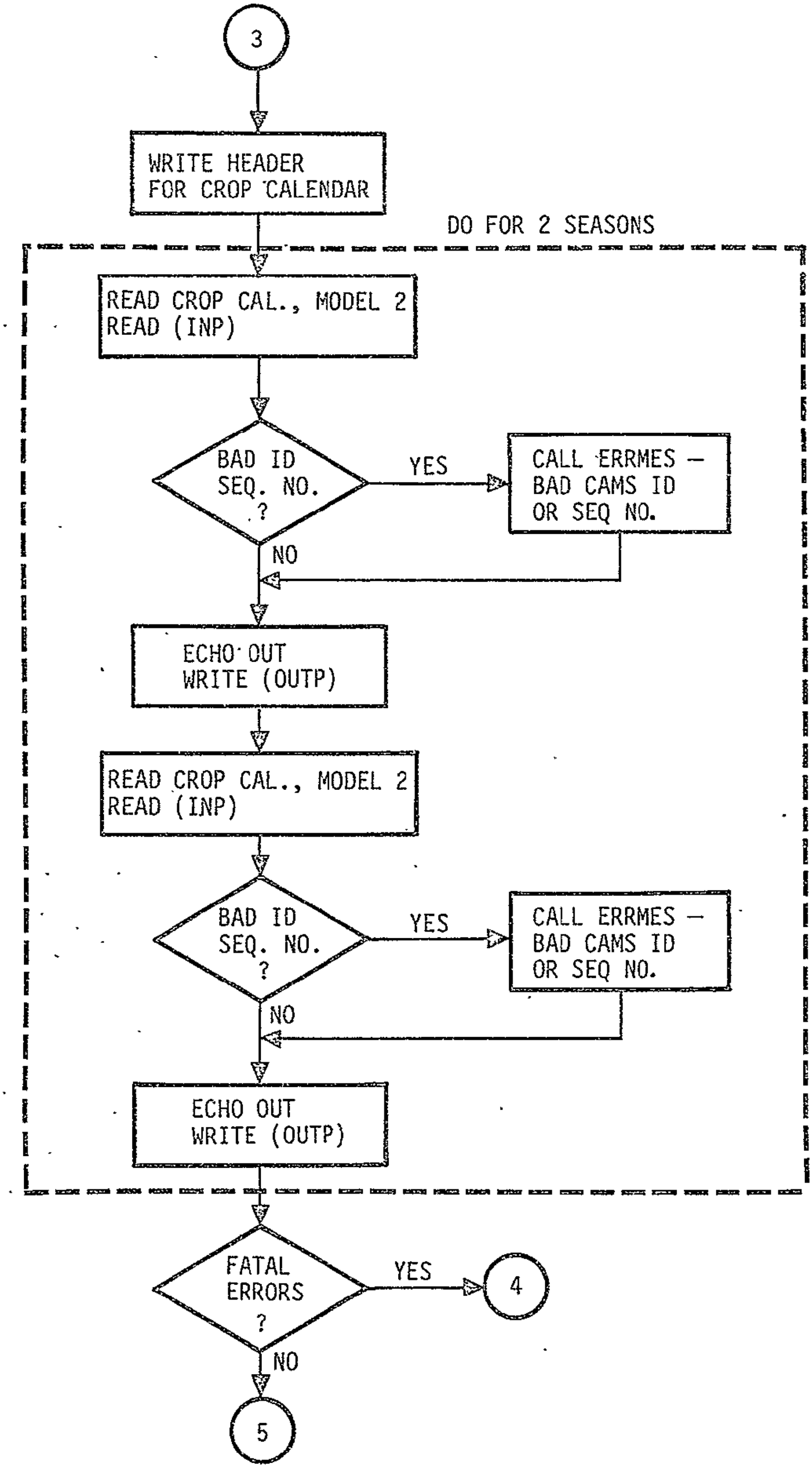
SUBROUTINE CAMSIN BLOCK DIAGRAM



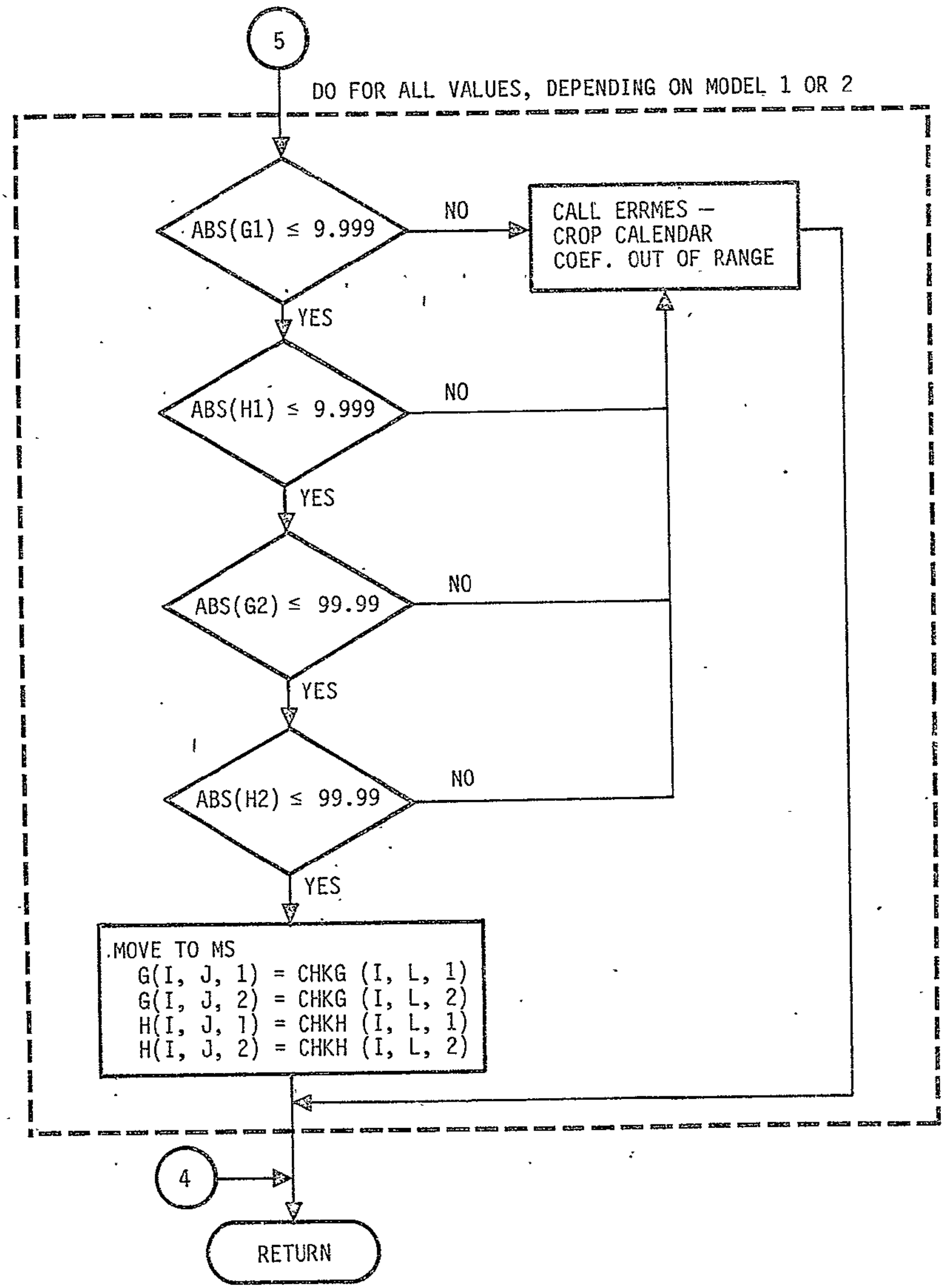
CAMSIN (CONT'D)



CAMSIN (CONT'D)



CAMSIN (CONT'D)



SUBROUTINE BETAD

Purpose: To compute a random number based on the Beta distribution or normal distribution, given a random number seed.

Input: No input from COMMON or files.

Output: No output to COMMON or files.

Linkage: CALL BETAD (SEED, XBAR, SIGMA, XI, IOPT, IER)

Input:

SEED	A double precision random number seed used to get a uniform random number P, $0 < P < 1$	
XBAR	Mean value $\overline{X}$, $0 \leq \overline{X} \leq 1$	
SIGMA	Standard deviation σ, $0 \leq \sigma$	
IOPT	= 0 use Beta distribution $\neq 0$ use normal distribution	

Output:

XI	Random number based on Beta or normal distribution X_i
IER	Error flag = 0 no errors = 1 XBAR not in range, $0 \leq \overline{X} \leq 1$ so was reset within subroutine = 2 SIGMA not in range, $0 \leq \sigma \leq \overline{X}\sqrt{\frac{1-\overline{X}}{\overline{X}+\epsilon}}$ so was reset within subroutine, $\epsilon = 10^{-4}$ = 3 Fatal error, XI could not be found within constraints of subroutine; e.g., within 35 iterations via the inverse incomplete beta function method
SEED	To be used for next call to BETAD (a double precision number)

Subroutines used:

CALL RDM1A (SEED, P) to get uniform random number P
SEED = double precision

CALL IBETAI (X, A, B, P, IER) to get incomplete beta function

Note: IBETAI is algorithm AS 63 Appl. Statist. (1973), Vol. 22, No. 3

SQRT (X)	squareroot
ALOG (X)	exponential
EXP (X)	natural logarithm
ALOG(X)	natural logarithm

Local variables:

A		First Beta parameter
B		Second Beta parameter
BP		Recalculated second Beta parameter
CHK		Normal distribution parameter
DIFF		Accuracy check
DIFF1		Check if XI close to 0
EP		10^{-4}, accuracy of answer
FLAG	INTEGER	Flag to signal XBAR > .5
H		Beta approx. parameter
I		Loop counter
K	REAL	2., method threshold constant
P		Output from RDM1A, f(X) for Beta function
PHI		Limit for iteration of P
PLO		Limit for iteration of P
PO		Output from IBETAI
R		88., Gamma constraint
RN		Normal distribution parameter
SIG		Stores SIGMA, or SIGMAL, for use in routine
SG		10^{-10} check on successive answers in loop
SIGMAL		Upper limit on SIGMA
SIGSQ		SIG * SIG, intermediate calculation
SIGT		Method threshold sigma
T		Normal distribution parameter
W		Beta approx. parameter
XAVG		XBAR, or 1 - XBAR if XBAR > .5
XHI		Limit for iteration of X
XLO		Limit for iteration of X
XSQ		XBAR * XBAR, intermediate calculation
Y		Beta approx. parameter
YP		Beta approx. parameter

BETAD Subroutine Equations

Equation set 1 - normal distribution parameters:

$$T = \begin{cases} \sqrt{\ln \dfrac{1}{P^2}} & 0 < P \leq .5 \\ \sqrt{\ln \dfrac{1}{(1-P)^2}} & .5 < P < 1 \end{cases}$$

$$CHK = T - \frac{2.30753 + .27061T}{1. + .99229T + .04481T^2}$$

$$RN = \begin{cases} -CHK & 0 < P \leq .5 \\ +CHK & .5 < P < 1 \end{cases}$$

Equation set 2 - SIGMA upper limit:

$$SIGMAL = XBAR \sqrt{\frac{1 - XBAR}{XBAR + EP}} \qquad EP = 10^{-4}$$

Equation set 3 - Beta function parameters:

$$A = \frac{XBAR^2 - XBAR\,(XBAR^2 + SIGMA^2)}{SIGMA^2}$$

$$B = \left(\frac{1 - XBAR}{XBAR}\right) A$$

Equation set 4 - method threshold:

$$SIGT = XBAR \sqrt{\frac{1 - XBAR}{XBAR + K}} \qquad K = 2$$

Equation set 5 - Beta approximation parameters:

$$YP = -RN$$

$$H = 2\left(\frac{1}{2A - 1} + \frac{1}{2B - 1}\right)^{-1}$$

$$Y = \frac{YP^2 - 3}{6}$$

$$W = \frac{YP\,(H + Y)^{1/2}}{H} - \left(\frac{1}{2B - 1} - \frac{1}{2A - 1}\right)\left(Y + \frac{5}{6} - \frac{2}{3H}\right)$$

Equation set 6 - XI for Beta approximation:

$$XI = \frac{A}{A + B \cdot e^{2W}} \qquad ABS\,(ALOG\,(B) + 2 * W) \leq 87$$

Equation set 7 - recompute A and B, Beta parameters:

$$B^1 = \frac{B}{A + B} \cdot (R - 1)$$

$$A^1 = \frac{BP}{B} \cdot A$$

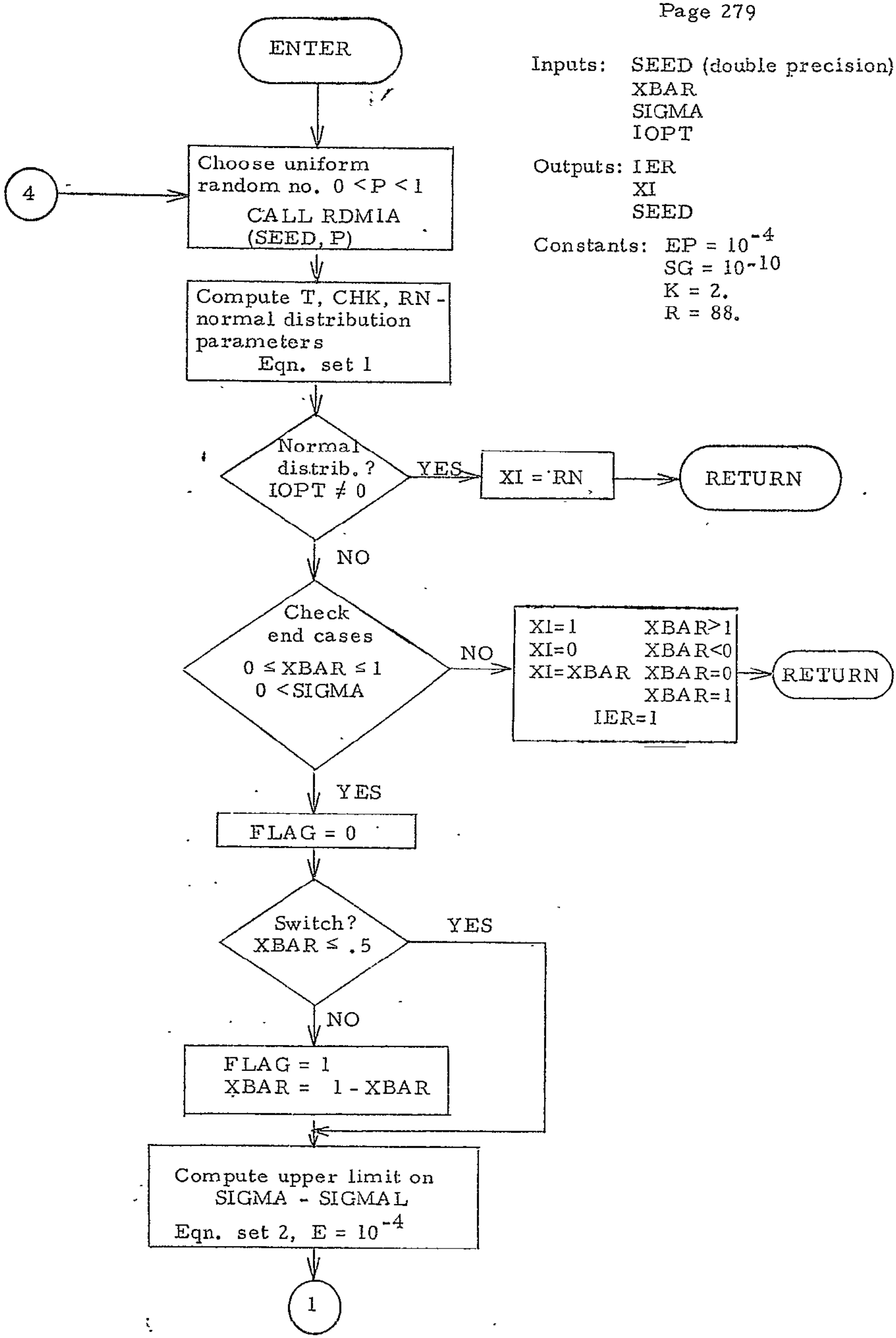
Equation set 8 -

$$XI = \frac{(XH1 + XL0)}{2.}$$

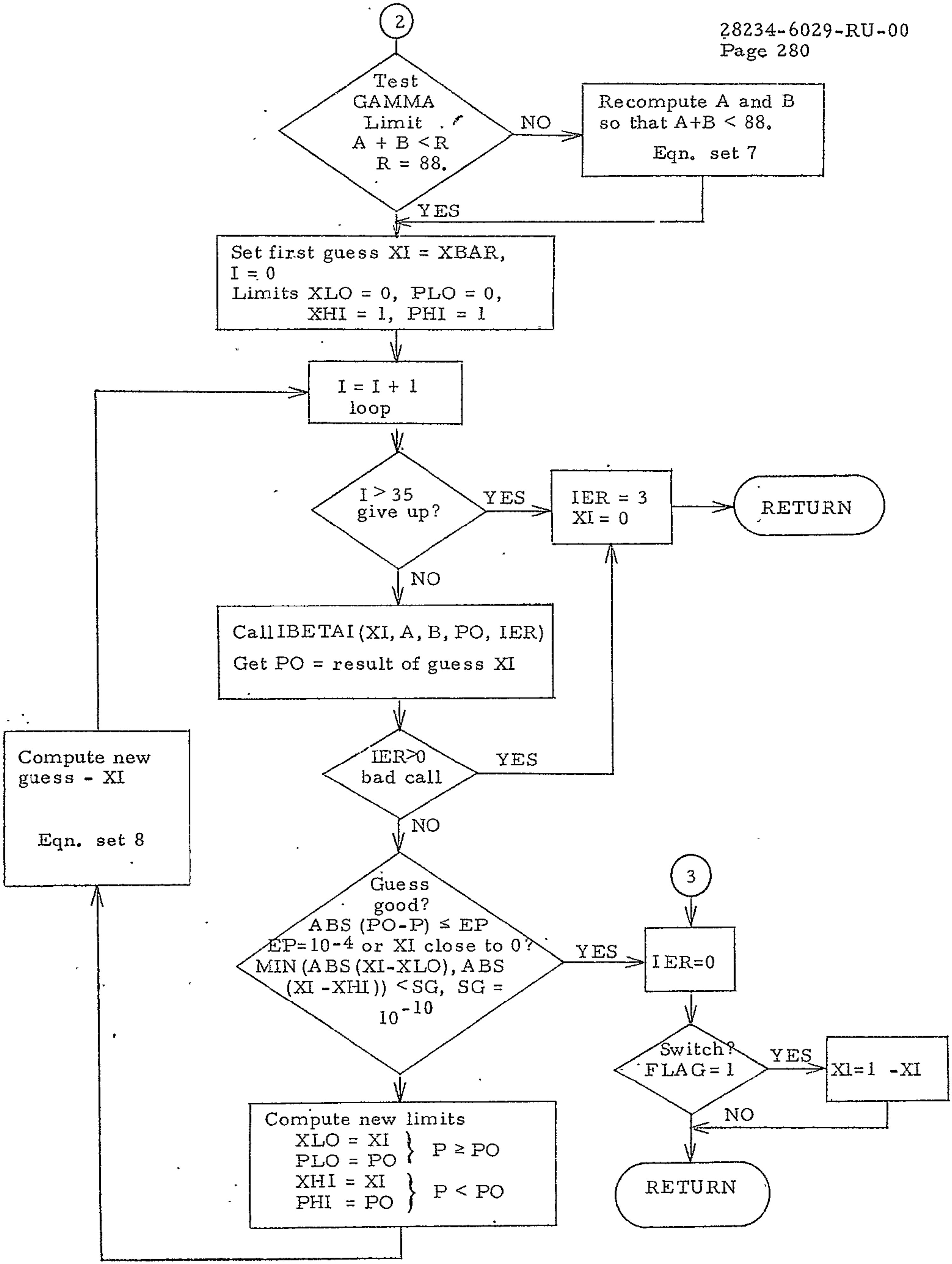
Inputs: SEED (double precision)
XBAR
SIGMA
IOPT

Outputs: IER
XI
SEED

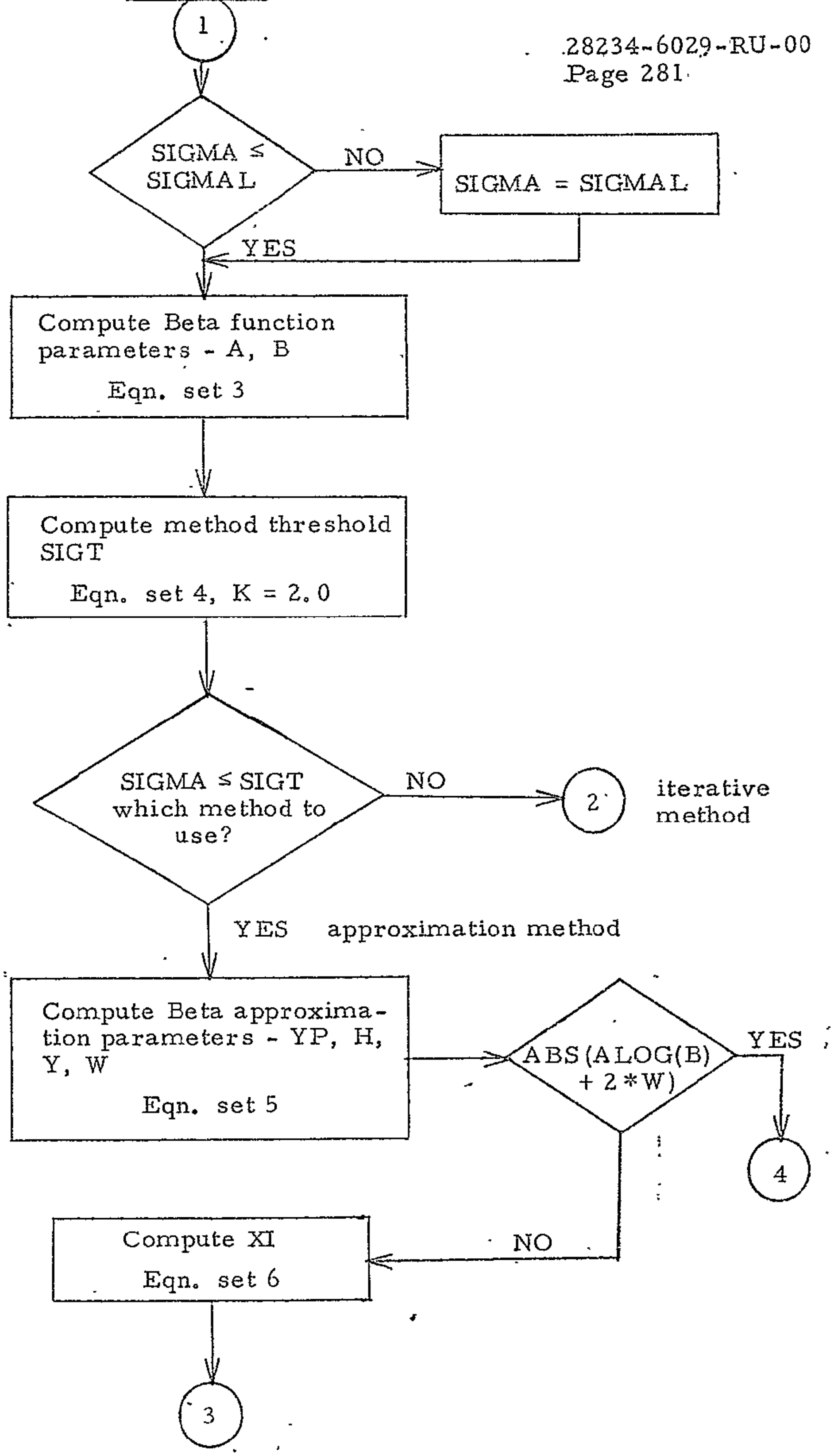
Constants: $EP = 10^{-4}$
$SG = 10^{-10}$
K = 2.
R = 88.



BETAD Subroutine Flowchart (Sheet 1 of 3)



BETAD Subroutine Flowchart (Sheet 3 of 3)



BETAD Subroutine Flowchart (Sheet 2 of 3)

SUBROUTINE CAMS

Purpose:

This is the driver for the CAMS module. It calls the appropriate subroutines to calculate intermediate error quantities and then combines them for the estimated proportion of wheat for each acquisition date for each segment. Depending on error bypass flags, certain error calculations may be bypassed. Model 2 is treated as a subset case of the more complex model 1, where the error factors for mixed and other fields are not computed. To do this, data from the input files and cards must be read in and stored depending on which model is used. Figure 6 of the CAMS Problem Description gives the flow of this subroutine.

Input:

/CAMSCM/	IMODEL IMULTI ISIGEX ISKIP ITMAX IREP IWIND IGROUP (3, 2, 15) MS (3, 2, 3) G (3, 2, 2) H (3, 2, 2)
/LEMCM/	ISEXT ISCC ICLASS ICAMS IACQ STARTZ ENDZ STARTR ENDR ICASE
/CNTRL/	PRINTF SEED(2) SEED(3) SEED(4)

Output:

/CNTRL/	SEED(2) SEED(3) SEED(4)
/ARGLST/	NFATAL
/PAGECM/	NPAGE NLINE
/STATS/	NREC(2) NREC(4) NREC(6) NREC(7) NCAMSR

CAMS output file (CAMSF) - see file descriptions.

CAMS printed report - see CAMS Problem Description, Figure 7.

CAMS error messages - see CAMS Problem Description, Section 5.3.

Linkage:

CALL CAMS - called from LEM program.

Subroutines Used:

CALL INPT (ISEG, IACQU, ICAMER, ICROP, ISIGEX, IMODEL, IPASS, IDONE, IEND) to read in input file records.

CALL INITI (ISEG, IACQ, ICAMER, ICROPW, ISIGEX, HEAD) to initialize input files, ready to read.

CALL CLASS (SEED(2), TYPE, WINDOW, M, BCC, SIGCC, XI) to compute classification error.

CALL MULTI (TYPE, SEASON, IWIN, M) to compute multi-temporal error.

CALL CROP (SEED(4), TYPE, SEASON, WINDOW, IFIRST, BCC, SIGCC, ITSEG) to compute crop calendar error.

CALL CORREL (ITMAX, ACQUIS, WINDOW, IUSE) to correlate training segment with ordinary segment.

CALL SIGEXT (SEED(3), TYPE, WINDOW, IUSE, ISIGEX, XI) to compute signature extension error.

CALL TSAVE (0, -1, IBAD) to close RA file TACQ.

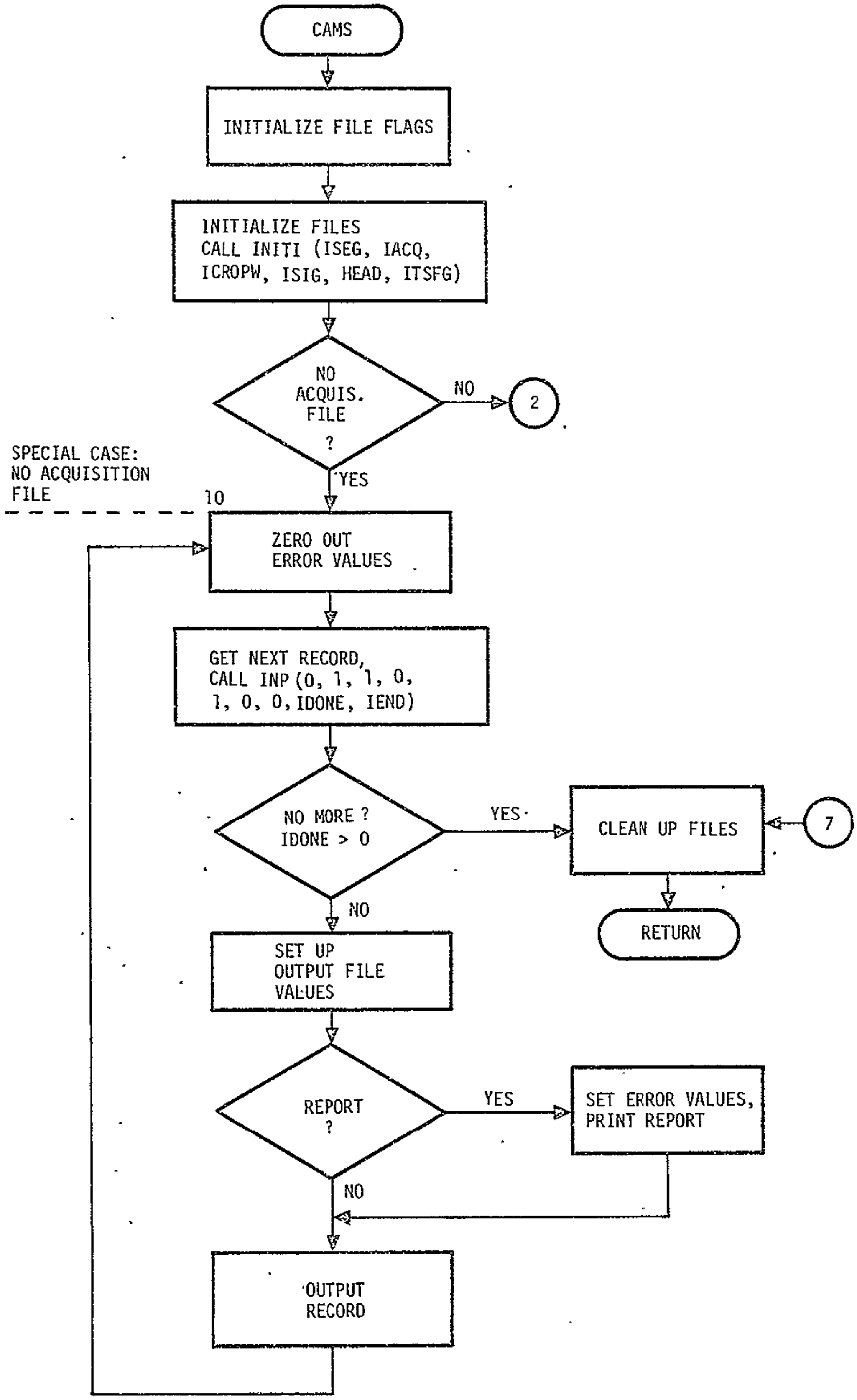
CALL REPORT (IPASS, IFIRST, IREP) to write report.

CALL EJECT (IND) to start report.

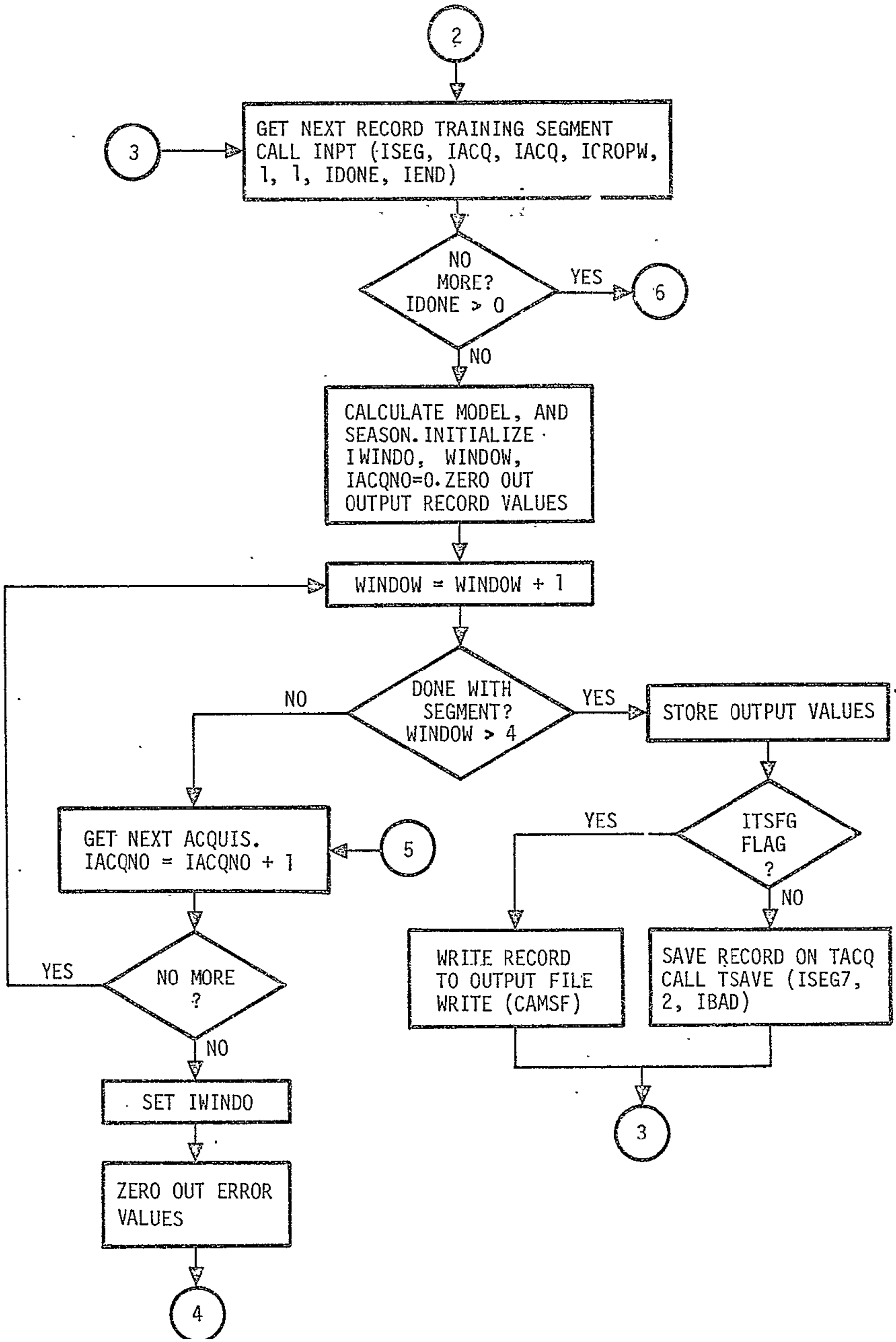
Local Variable Description:

WHE	integer, =1, wheat component
MIX	integer, =2, mixed component
OTH	integer, =3, other component
ERR	10^{-6} to keep from dividing by zero
IFIRST	flag for report, =1 if first acquisition for segment, >1 otherwise
SEASON	integer, which type wheat, 1 = winter, 2 = spring
WINDOW	integer, which window acquisition in, 1-4
TYPE	integer, which component (1 = wheat, 2 = mixed, 3 = other)
MODEL	integer, how many iterations to do (1 = model 2, 3 = model 1)
HEAD(4, 4)	holds window titles from INITI subroutine
XI(3)	holds total error from CLASS or SIGEXT
M(3)	holds multi-temporal error from MULTI
BCC(3)	holds crop calendar bias from CROP
SIGCC(3)	holds crop calendar sigma from CROP
P(3)	holds proportions, wheat, mixed, other
IWINDO(4)	flags for MULTI, = 0 no acquisition in window = 1 at least 1 acquisition in window
ICROPW	flag for INITI and INPT, CROPW file bypass
ISIG	flag for INITI and INPT, SIGEXT file bypass
ISEG	flag for INPT, SEGTRU file bypass
IACQNO	what acquisition no. on, 1-25
IDONE	output of INPT
IEND	output of INPT
IUSE	output of CORREL
I, J	indexes for DO loops
IFILL	filler for trailer record, output file
ITOT	no. of words of filler IFILL
ZERO1 ZERO2	used to prevent divides by 0
ZZZZ	contains ZZZZ for trailer record

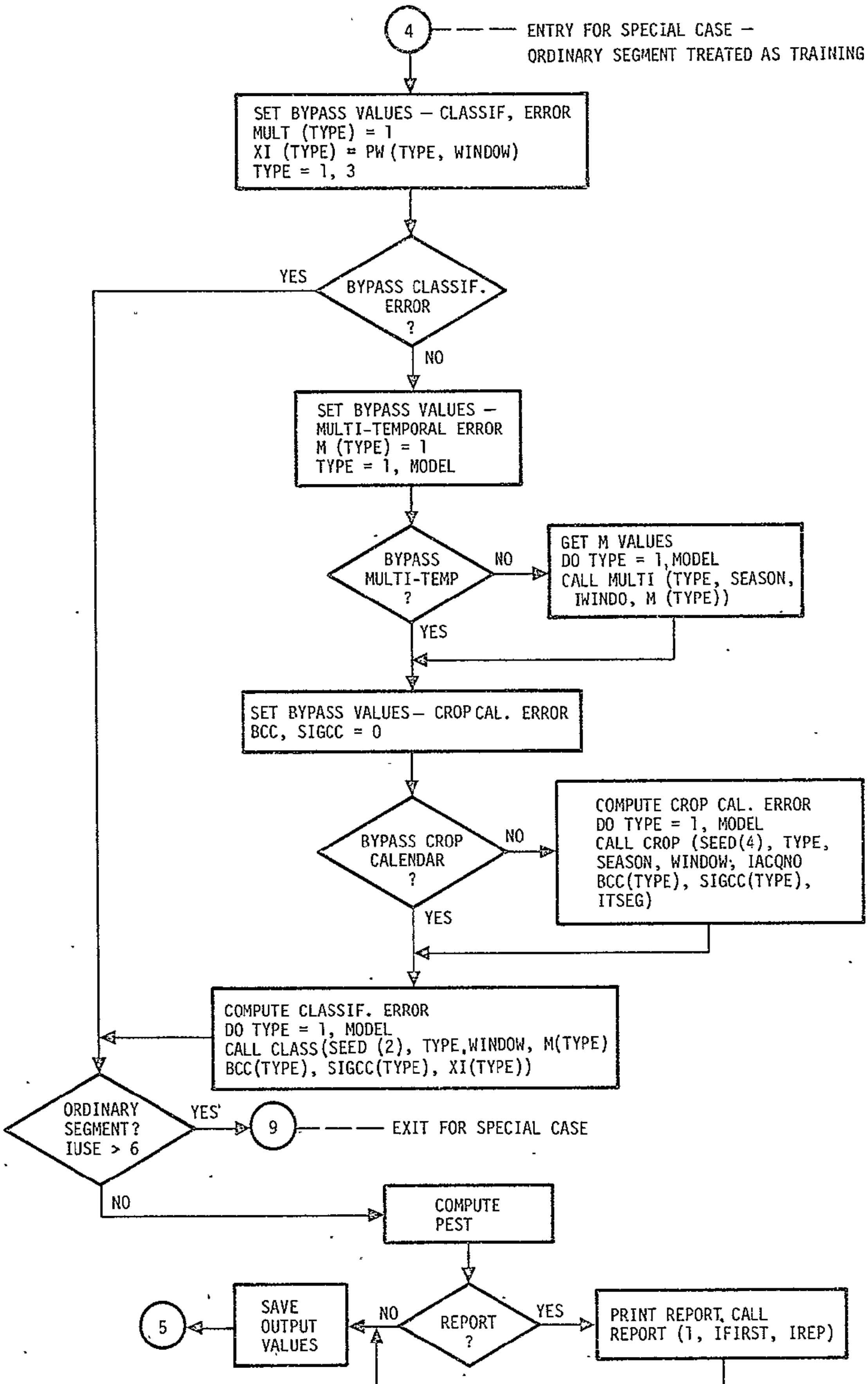
SUBROUTINE CAMS BLOCK DIAGRAM



PASS 1 — TRAINING SEGMENTS

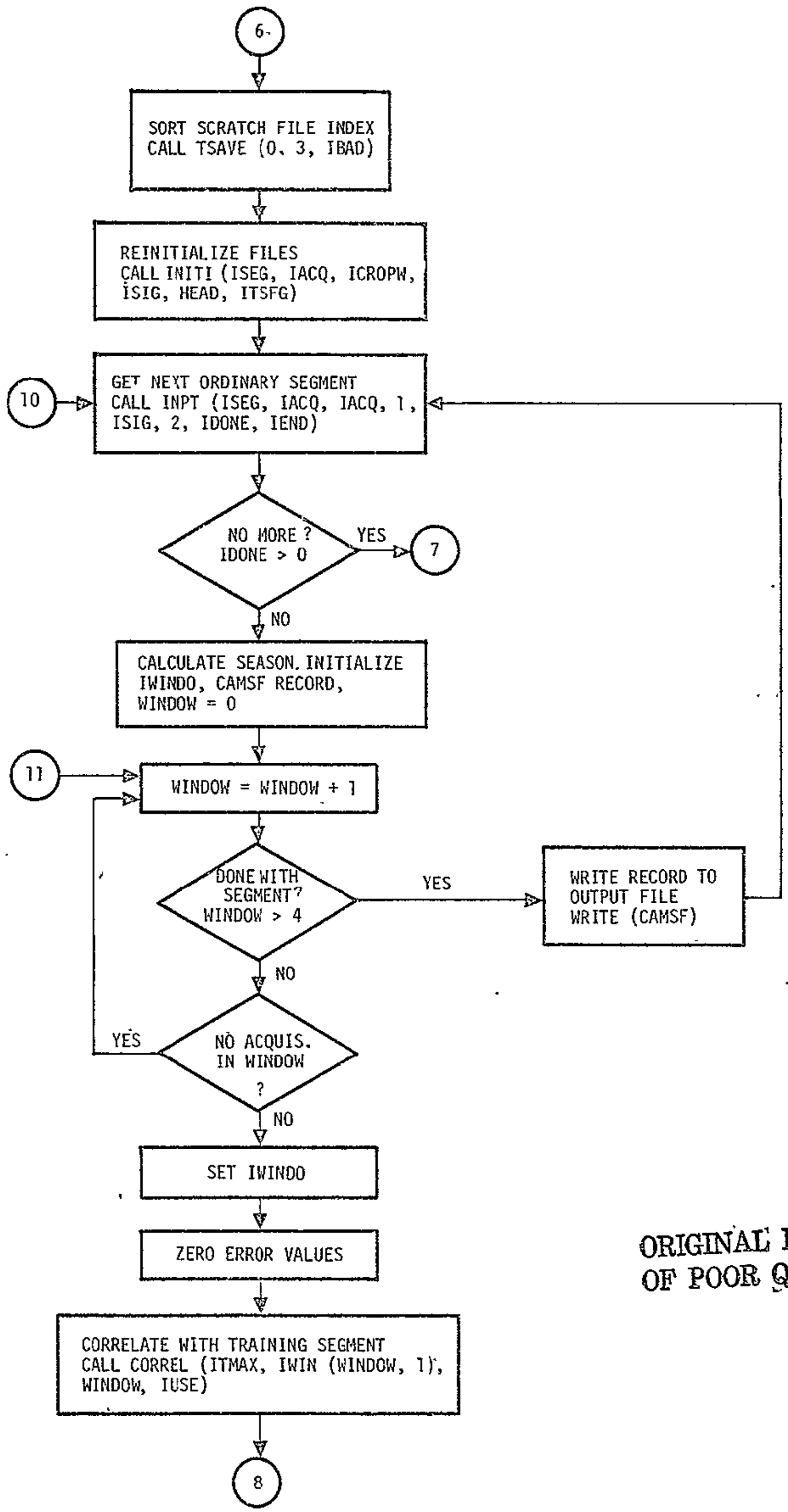


CAMS (CONT'D)


4
ENTRY FOR SPECIAL CASE — ORDINARY SEGMENT TREATED AS TRAINING
SET BYPASS VALUES — CLASSIF, ERROR
MULT (TYPE) = 1
XI (TYPE) = PW (TYPE, WINDOW)
TYPE = 1, 3
BYPASS CLASSIF. ERROR ?
YES
NO
SET BYPASS VALUES — MULTI-TEMPORAL ERROR
M (TYPE) = 1
TYPE = 1, MODEL
BYPASS MULTI-TEMP ?
NO
YES
GET M VALUES
DO TYPE = 1, MODEL
CALL MULTI (TYPE, SEASON, IWINDO, M (TYPE))
SET BYPASS VALUES — CROP CAL. ERROR
BCC, SIGCC = 0
BYPASS CROP CALENDAR ?
NO
YES
COMPUTE CROP CAL. ERROR
DO TYPE = 1, MODEL
CALL CROP (SEED(4), TYPE, SEASON, WINDOW, IACQNO
BCC(TYPE), SIGCC(TYPE), ITSEG)
COMPUTE CLASSIF. ERROR
DO TYPE = 1, MODEL
CALL CLASS (SEED (2), TYPE, WINDOW, M(TYPE)
BCC(TYPE), SIGCC(TYPE), XI(TYPE))
ORDINARY SEGMENT? IUSE > 6
YES
9
EXIT FOR SPECIAL CASE
NO
COMPUTE PEST
REPORT ?
YES
PRINT REPORT, CALL REPORT (1, IFIRST, IREP)
NO
SAVE OUTPUT VALUES
5

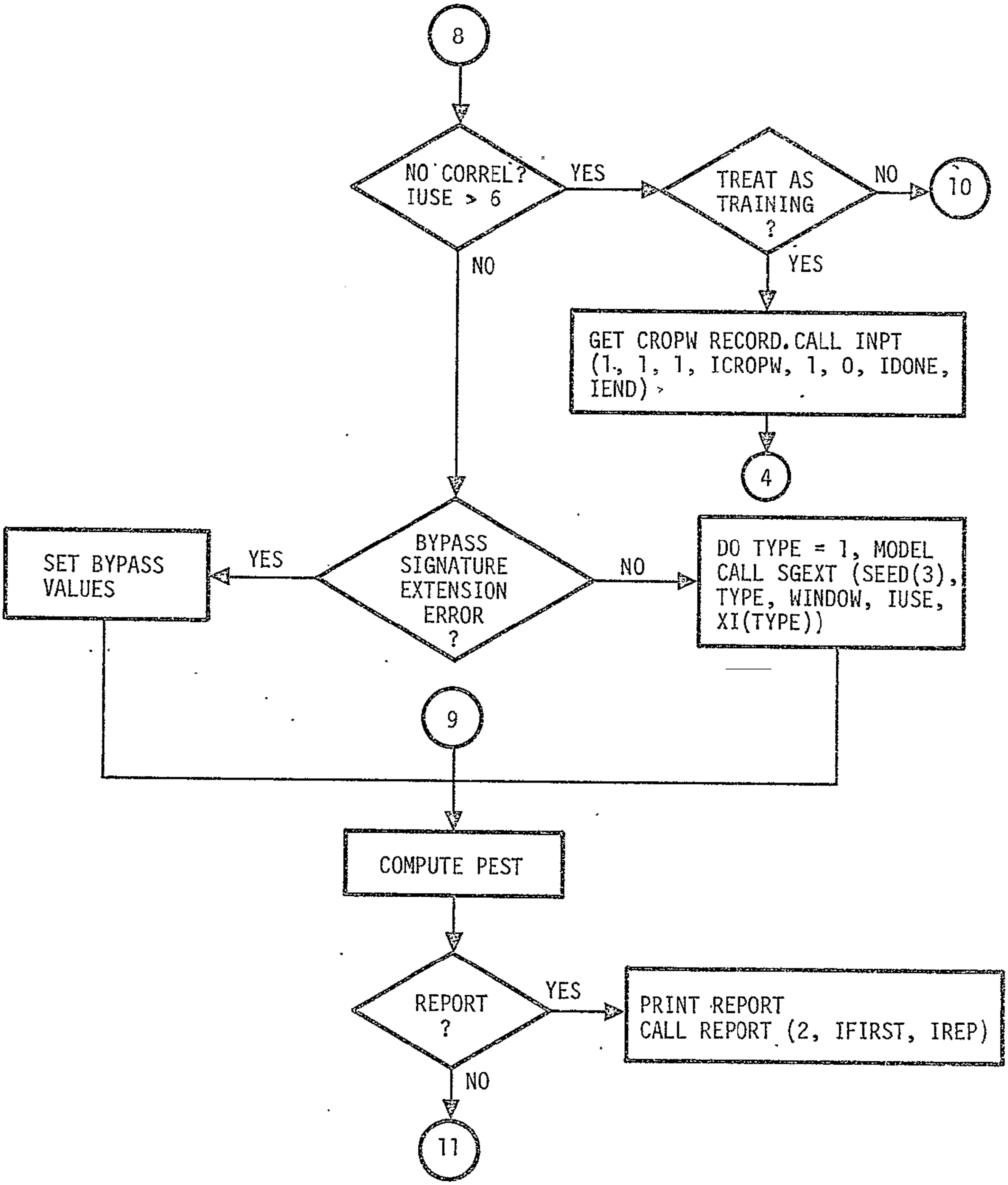
CAMS (CONT'D)

PASS 2 — ORDINARY SEGMENTS



ORIGINAL PAGE IS
OF POOR QUALITY

CAMS (CONT'D)


8
NO CORREL? IUSE > 6
YES
TREAT AS TRAINING ?
NO
10
YES
NO
GET CROPW RECORD. CALL INPT (1, 1, 1, ICROPW, 1, 0, IDONE, IEND)
4
SET BYPASS VALUES
YES
BYPASS SIGNATURE EXTENSION ERROR ?
NO
DO TYPE = 1, MODEL CALL SGEXT (SEED(3), TYPE, WINDOW, IUSE, XI(TYPE))
9
COMPUTE PEST
REPORT ?
YES
PRINT REPORT CALL REPORT (2, IFIRST, IREP)
NO
11

SUBROUTINE REPORT

Purpose:

This subroutine outputs the printed report for CAMS. On option, the breakdown of error values can be omitted. REPORT is organized into two passes -- training segments are listed first, then ordinary segments. The special case, no acquisition file, is handled separately. For model 2, with fewer values, zeros appear in the irrelevant fields. For the special case of an ordinary segment acquisition with no training segment correlation, the data is flagged.

Input:

/ERROR/	TITLE(4) IDATE PESTIM TOT ALOCAL ERTOT(3) ERBIAS(3) ERRAND(3) CLTOT(3) CLBIAS(3) CLRAND(3) CROPF CROPD SIGZ(3, 2) MULT(3) TID TRAINA TRAIND
/SEGTRU/	COUN4 IREG4 IZONE4 ISTR4 ISUB4 ISEG4 PT(1)
/FILES/	CROPW ACQUIS CAMS CAMERR SIGEXT SEGTRU INP OUTP TACQ LCAMSF

Output:

Printed report only. See CAMS Problem Description, Figure 7.

Linkage:

CALL REPORT (IPASS, IFIRST, IREP)

Inputs:	IPASS	=0	special case - no acquisition file
		=1	training segment pass
		=2	ordinary segment pass
	IFIRST	=1	first acquisition for segment
		>1	not first acquisition for segment
	IREP		report flag from CAMS control card
		=1	no error breakdown, just estimates
		=0	print error breakdown report too

Subroutines Used:

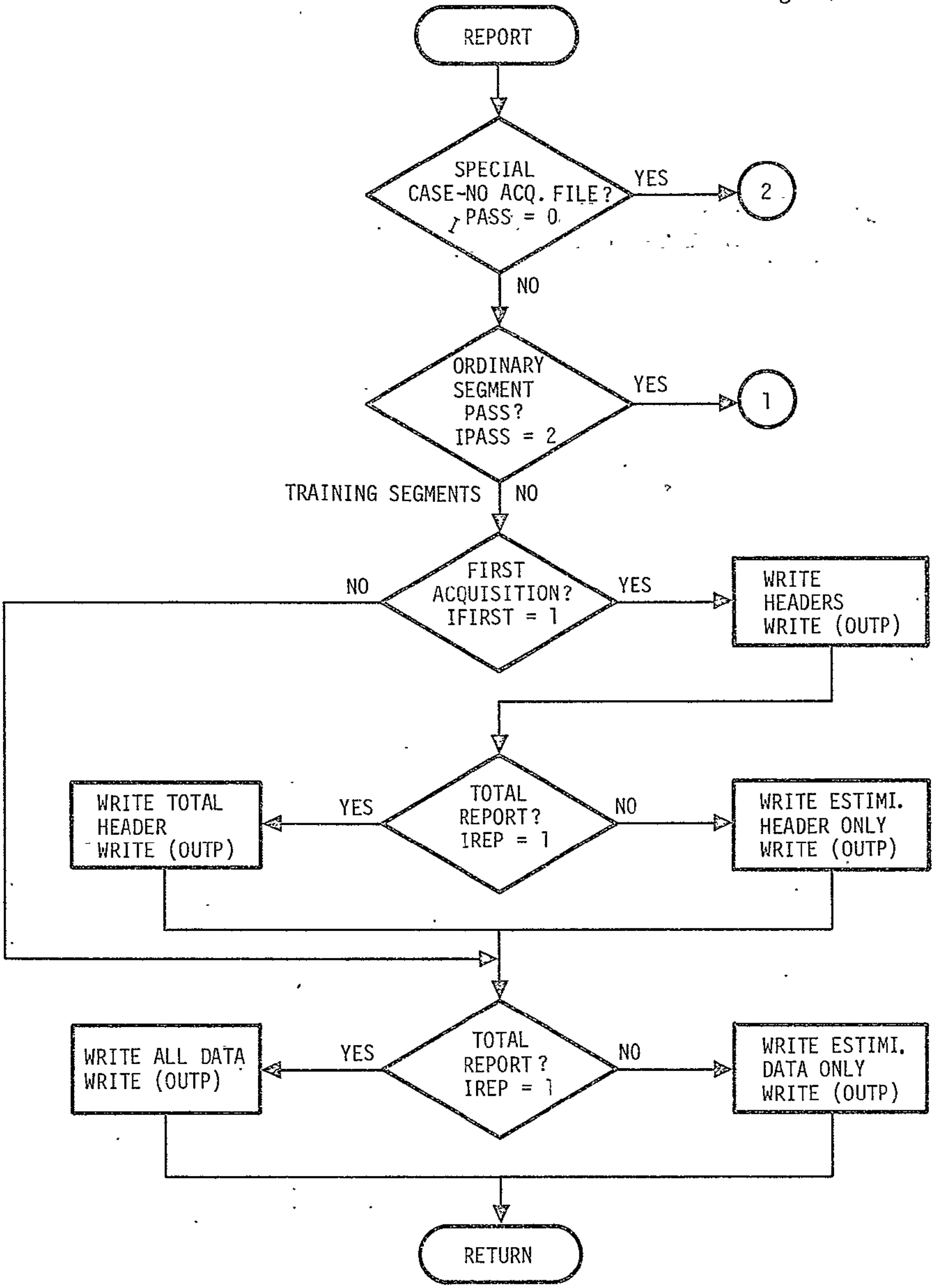
CALL PAGER (IND) to print headings.

CALL FZULU (DATE, IOUT) to get calendar date from Zulu date.

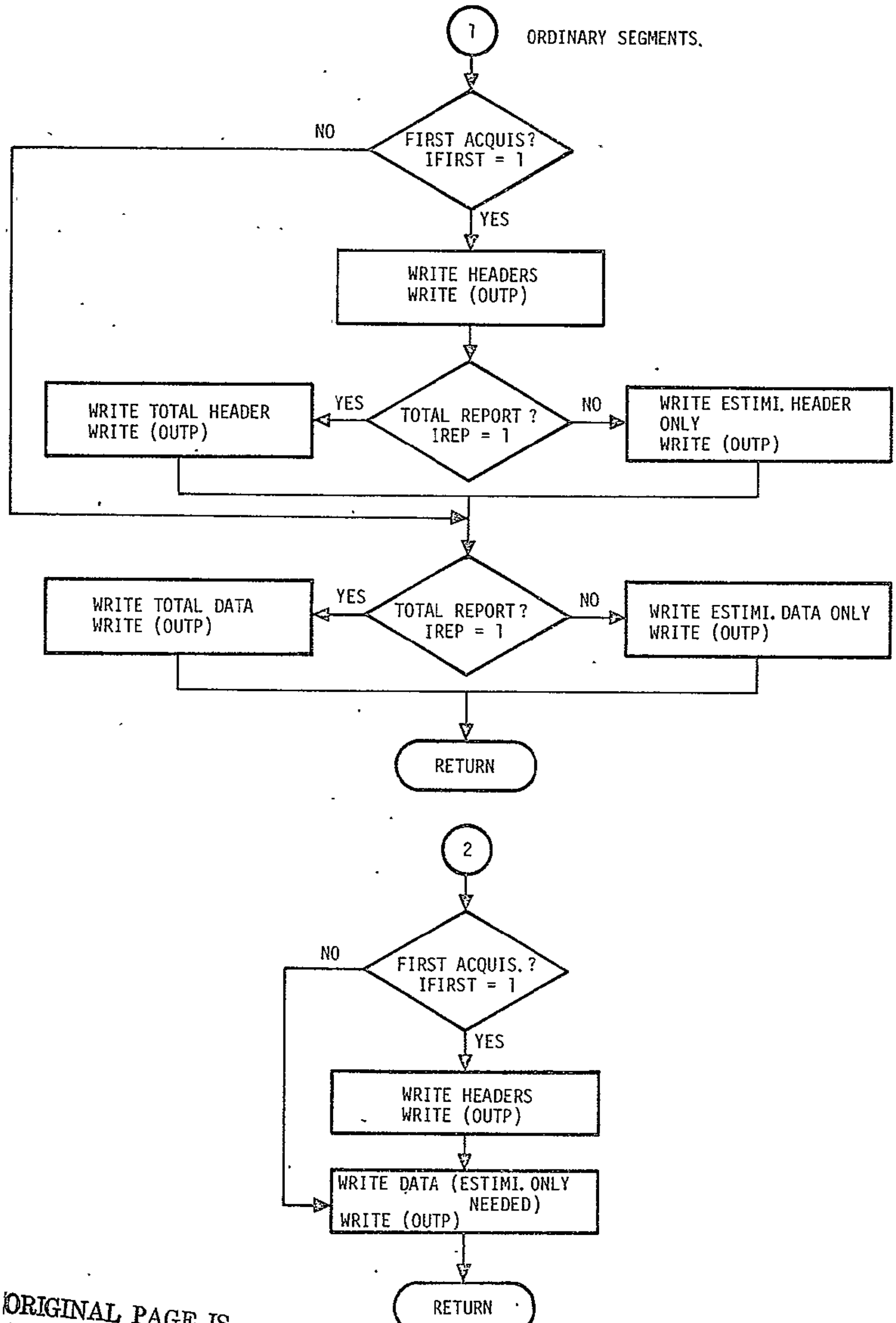
Local Variable Description:

IOUT(3) holds calendar date from FZULU subroutine

SUBROUTINE REPORT BLOCK DIAGRAM



REPORT (CONT'D)



SUBROUTINE INITI

Purpose:

This subroutine initializes the input and output files, plus the intermediate direct access file, for CAMS. It finds the first record to process on the key file SEGTRU, passes over the headers of the other input files, opens the direct access file, and outputs the header on the output CAMSF file. Depending on which error conditions are bypassed, some steps may be bypassed.

Input:

/LEMCM/	STARTR STARTZ	starting region and zone, integers
	IACQ ISCC ICLASS ISEXT ICAMS	for output file header
	ICASE	case no. of output file
/FILES/	SEGTRU ACQUIS TACQ CAMERR CROPW SIGEXT CAMSF	logical file unit nos., integers
	LCAMSF	length of output record
/CAMSCM/	IMODEL	model no. (1 or 2)
	IMULTI	for output file header

See also linkage.

Output:

/INDX/	INDEX	index for RA scratch file TACQ
/ARGLST/	NERRS	error count passed back from ERRMES subroutine
	NFATAL NPERRS	

See also linkage.

Linkage:

CALL INITI (ISEG, IACQU, ICAMER, ICROPW, ISGEX, HEAD, ITSFG)

Inputs:	ISEG			
	IACQU			
	ICAMER	flags for input files -	0	read file
	ICROPW		>0	skip reading file
	ISGEX			

Outputs: HEAD(4, 4) headings for the four crop windows from ACQUIS file unless bypassed, then defaults to
****WINDOW 1****
****WINDOW 2****
****WINDOW 3****
****WINDOW 4****

ITSFG flag if all training segments
=0 all training
≠0 training and ordinary

Subroutines Used:

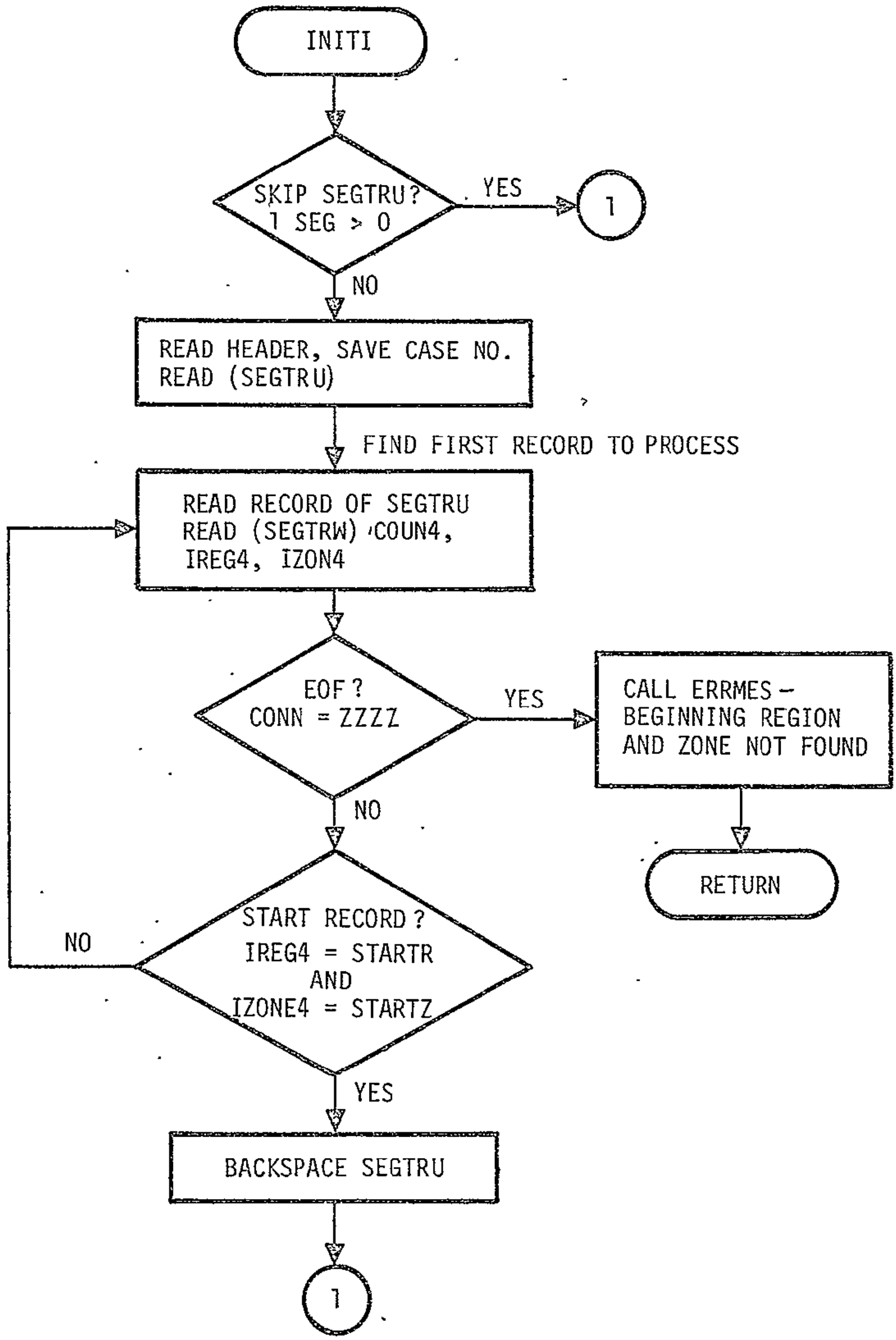
CALL ERRMES (4HCAMS, 4HINIT, 1, 1) to report error message.

CALL TSAVE (0, 0, IBAD) to open scratch file TACQ.

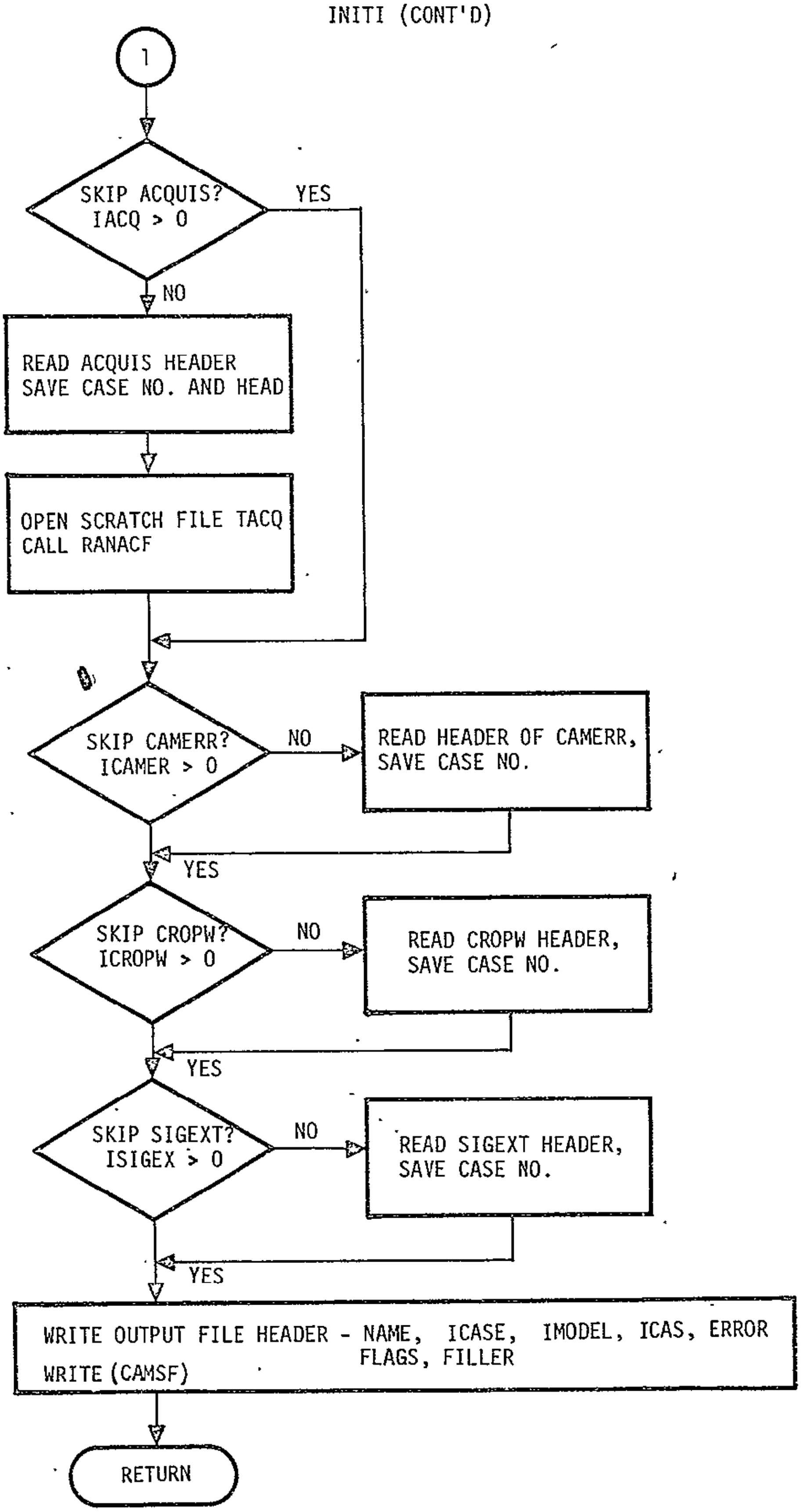
Local Variable Description:

NAME(2)	name of output file	
ICAS(5)	case nos. of input files	
IFILL	filler, = 0	
/ARGLST/	NARG	no. of arguments in error routine list
/SEGTRU/	COUN4	from SEGTRU input file record
	IREG4	
	IZON4	
ISKP	to skip over words	
ITOT	no. of filler words on header output file record	

SUBROUTINE INITI BLOCK DIAGRAM



INITI (CONT'D)


1
SKIP ACQUIS?
IACQ > 0
YES
NO
READ ACQUIS HEADER
SAVE CASE NO. AND HEAD
OPEN SCRATCH FILE TACQ
CALL RANACF
SKIP CAMERR?
ICAMER > 0
NO
READ HEADER OF CAMERR,
SAVE CASE NO.
YES
SKIP CROPW?
ICROPW > 0
NO
READ CROPW HEADER,
SAVE CASE NO.
YES
SKIP SIGEXT?
ISIGEX > 0
NO
READ SIGEXT HEADER,
SAVE CASE NO.
YES
WRITE OUTPUT FILE HEADER - NAME, ICASE, IMODEL, ICAS, ERROR FLAGS, FILLER
WRITE (CAMSF)
RETURN

SUBROUTINE MULTI

Purpose:

This subroutine returns the multi-temporal sampling error factor from the card input matrix, depending on the kind of wheat (winter, spring), the type (wheat, mixed, other), and which windows have had acquisitions. Model 2 is treated as though kind of wheat is wheat only. A value for the error report is saved.

Input:

/CAMSCM/	IGROUP(3, 2, 15) MS(3, 2, 3)	from card input

See also linkage.

Output:

/ERROR/	MULT(3)	for error report

See also linkage.

Linkage:

CALL MULTI (TYPE, SEASON, IWIN, M)

Inputs:

TYPE	integer, which component of mixed crops (1 = wheat, 2 = mixed, 3 = other; for model 2, 1 = wheat only used)
SEASON	integer, which kind of wheat (1 = winter, 2 = spring)
IWIN(4)	integer flag for each window, if had acquisition = 1, if not = 0

Outputs:

M	multi-temporal error factor

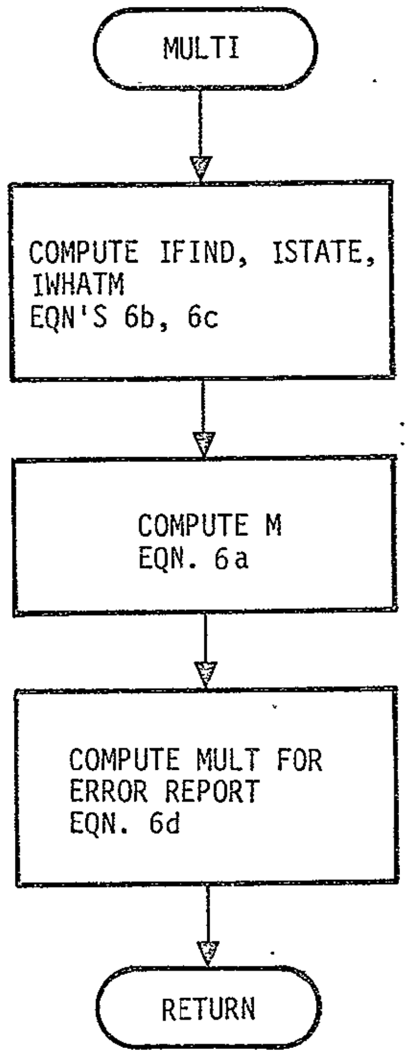
Subroutines Used:

None.

Local Variable Description:

IFIND	calculates which grouping to use, given which windows have acquisitions. It works as though INDEX(IFIND) really were INDEX (IWIN(1) + 1, IWIN(2) + 1, IWIN(3) + 1, IWIN(4) + 1), a 4-dimensional array, where, for example: INDEX(2, 1, 1, 1) gives which group to use if only acquisitions for first window; INDEX(1, 2, 1, 1) group if only acquisitions for second window.
INDEX(16)	index to which group to use
ISTATE	which group to use, equation B4-6b
IWHATM	which M to use, equation B4-6c

SUBROUTINE MULTI BLOCK DIAGRAM



ALL EQUATIONS FROM CAMS PROBLEM DESCRIPTION, SECTION 3.3,B

SUBROUTINE SGEXT

Purpose:

This subroutine calculates the signature extension error. It uses the bias and sigma from the SIGEXT input file to generate a random number from a beta distribution. Quantities for the error report are also computed and saved if needed.

Input:

/CAMERR/	PW(3, 4)	probabilities
/TRAINS/	TM(TYPE) TB(TYPE) TV(TYPE) TERTOT(TYPE)	error quantities of training segment
/SIGEXT/	ZB(3, 2) ZSIG(3, 2, 6)	bias and sigma for signature extension error
/CNTRL/	PRINTF	print flag, integer
/CAMSCM/	IREP ISIGEX	print flag for error report

See also linkage.

Output:

/ERROR/	ERTOT(TYPE) SIGZ(TYPE, 1) SIGZ(TYPE, 2) X(TYPE, 1) X(TYPE, 2)	error report quantities

See also linkage.

Linkage:

CALL SIGEXT (SEED(3), TYPE, WINDOW, IUSE, XI)

Inputs:

SEED(3)	double precision random no. from beta distribution for signature extension error
TYPE	integer, which component of mixed crops (1 = wheat, 2 = mixed, 3 = other; model 2 uses 1 = wheat only)
WINDOW	integer, which kind of wheat (1 = winter, 2 = spring)
IUSE	from CORREL subroutine, which training segment using

Outputs:

XI	total error factor
SEED(3)	double precision random no. seed after use

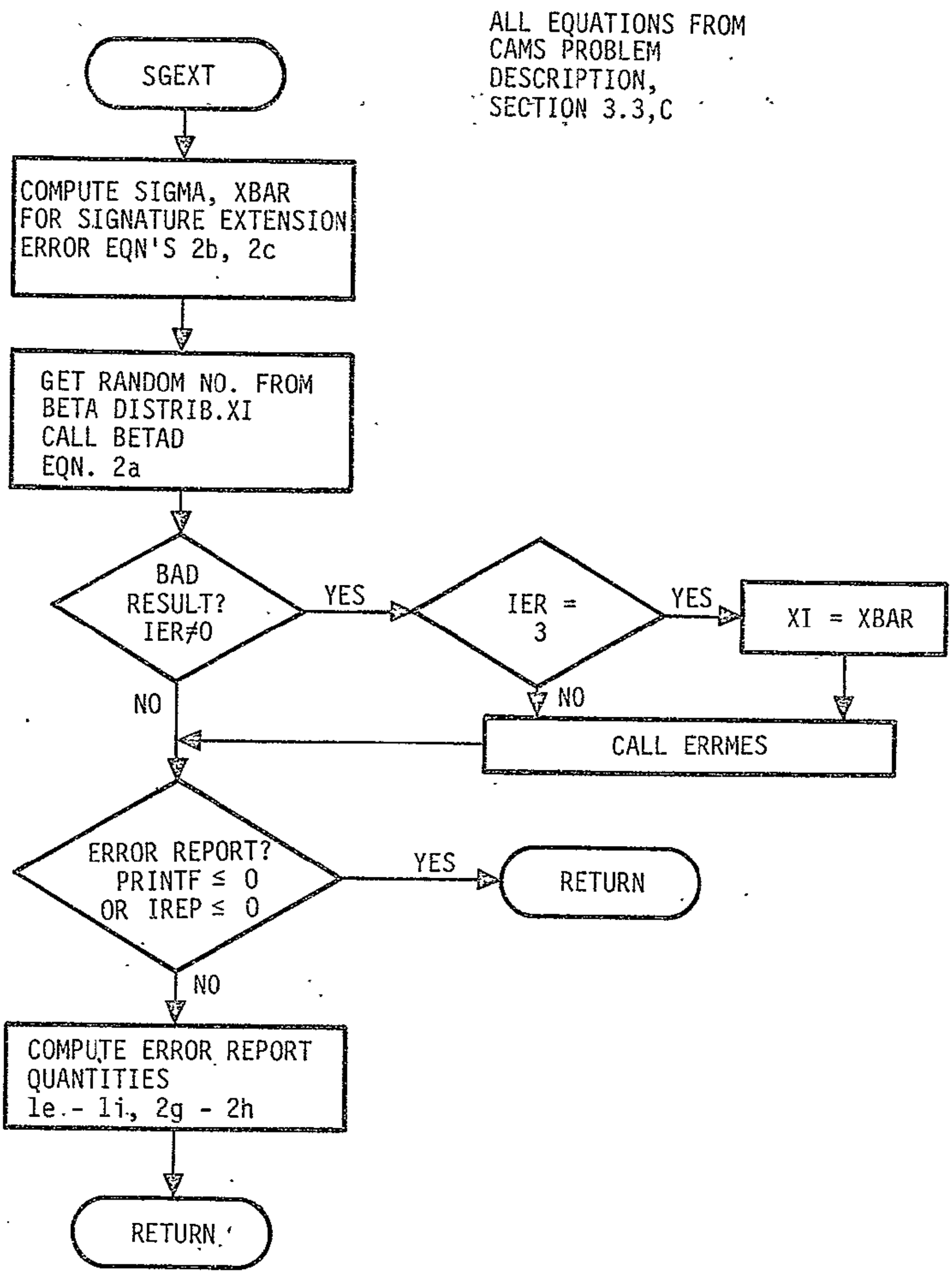
Subroutines Used:

CALL BETAD (SEED(3), XBAR, SIGMA, XI, 0, IER) to get random no. from beta distribution

Local Variation Description:

SIGMA, XBAR	sigma and average X for signature extension error, equations C.2b, 2c
IER	error flag from BETAD subroutine

SUBROUTINE SGEXT BLOCK DIAGRAM



SUBROUTINE CROP

Purpose:

This subroutine calculates the crop calendar error. It gets a random number from a normal distribution, the difference between the window start for the zone (from the CROPW file) and the actual window start for the segment. It then computes a bias and sigma, including this random number and quadratic function coefficients inputted from control cards. It also computes and saves values for the error report.

Input:

/CAMSCM/	G(3, 2, 2) H(3, 2, 2)	quadratic function coefficients
/CROPW/	START(2, 4) END(2, 4) SD(2) ERR(2, 5)	true start and end times for windows on zone level plus standard deviation and bias for actual start of segment within zone

Also input parameters.

Output:

/ERROR/	CROPF CROPD	error factors for error report

Also output parameters.

Linkage:

CALL CROP (SEED(4), TYPE, SEASON, WINDOW, IFIRST, BCC,
SIGCC, ITSEG)

Inputs:

SEED(4)	random no. seed for random number from normal distribution (double precision)
TYPE	which component of mixed crops, integer (1 = wheat, 2 = mixed, 3 = other; wheat only for model 2)
SEASON	which kind of wheat (winter, spring), integer
WINDOW	which window acquisition date in (1, 2, 3, or 4), integer
IFIRST	flag for first acquisition window, =1 for first >1 for rest

Outputs:

ITSEG	Δt saved, since only calculated for first acquisition in each window (equation 5d)
BCC	bias of crop calendar error
SIGCC	sigma of crop calendar error
SEED(4)	random number seed after used to calculate DELTA, double precision (used only for first acquisition in each window)

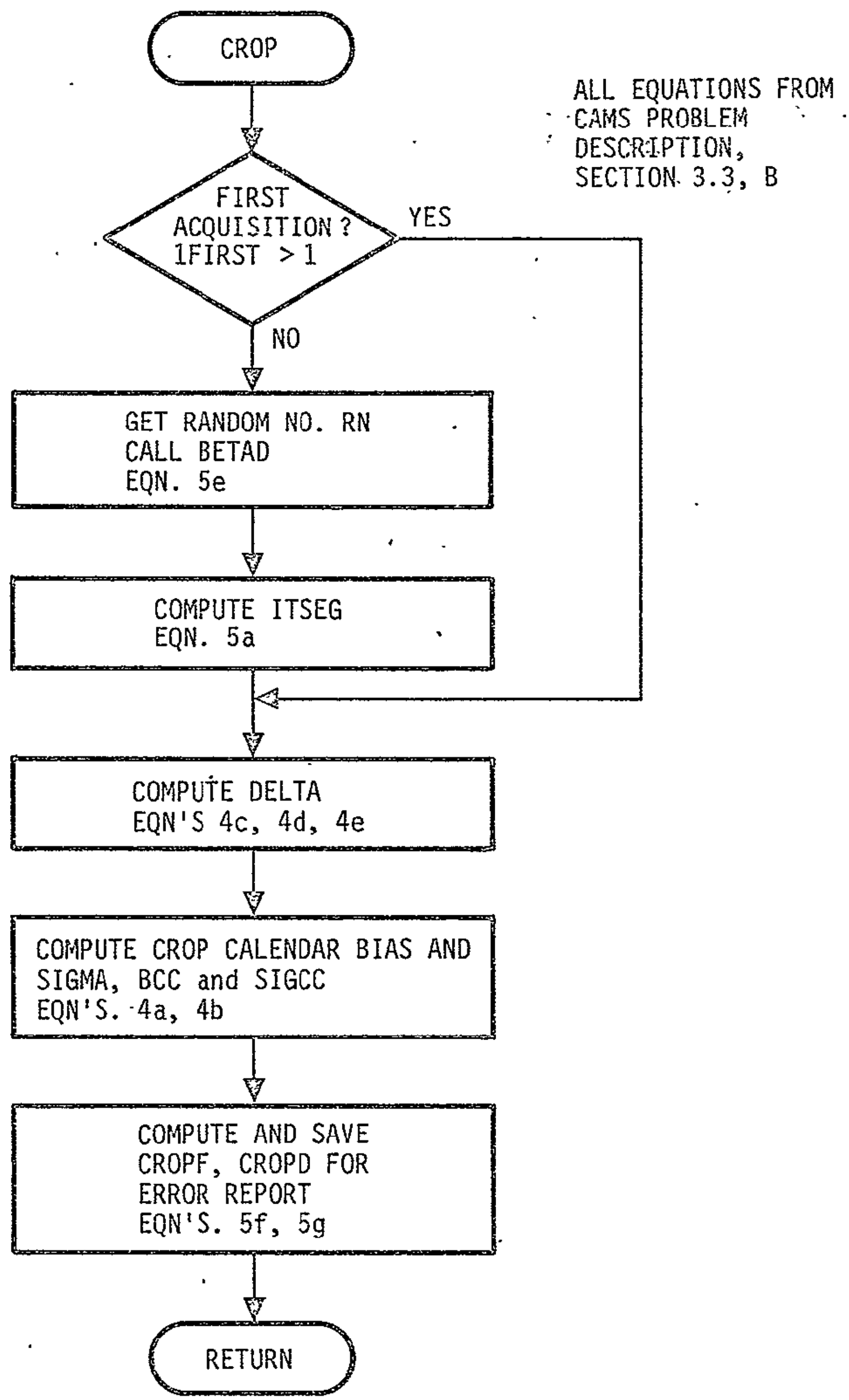
Subroutines Used:

CALL BETAD (SEED(4), 0, 0, RN, 1, IER) for returns RN, a random number from a normal distribution, from SEED(4).

Local Variable Description:

RN	random no. from normal distribution
BGNSEG, ENDSEG, TSTART	see equations B3, 5b-5d, CAMS Problem Description
IER	from call to BETAD; always returns 0

SUBROUTINE CROP BLOCK DIAGRAM



SUBROUTINE TSAVE

Purpose:

This subroutine handles the I/O for the scratch DA file TACQ.

Input:

/TRAINS/	COUN7 IREG7 IZONE7 ISTRA7 ISUB7 ISEG7 ITWIN(4, 25) ITTOT TMM(3, 4, 25) TBB(3, 4, 25) TV(3, 4, 25) TPTRUE TIZULU TPEST TPERR	1 record for file TACQ - see COMMON or file descriptions ISEGT = ID number = key to file
	TERTOT(3)	calculated from TM, TB, and TV
	TM(3) TV(3) TB(3)	
/INDX/	INDEX (2000)	index to TACQ (needed for CDC version)
	IPOINT (2001)	index to IPNT2 (contains segment ID's)
	IPNT2 (2001)	index to INDEX (after sort, is not necessarily the same as INDEX)
	IPEND	pointer to last word of IPOINT, IPNT2
	IPIN	pointer to last record read in (so no reread)
/FILES/	TACQ	logical unit no.

See also linkage.

Output:

/TRAINS/	same as INPUT description - depends on whether TACQ is being written to or read from
/INDX/	same as INPUT description
/ARGLST/NFATAL	TSAVE has fatal error if too many records

See also linkage.

Linkage:

CALL TSAVE (ISEG, IOPT, IBAD)

Inputs:	ISEG	segment ID no. of segment being written or read	
	IOPT	= 0	open file
		= 1	read file
		= 2	write file
		=-1	close file
		= 3	sort index, after all writes
	IBAD	= 0	segment found and read
		= 1	segment not found to read (may happen when start and end zones specified)

Subroutines Used:

CALL RANACF (IFILE, IREC, BUF, N, IX, L, IOPT) to use CDC RA routines.

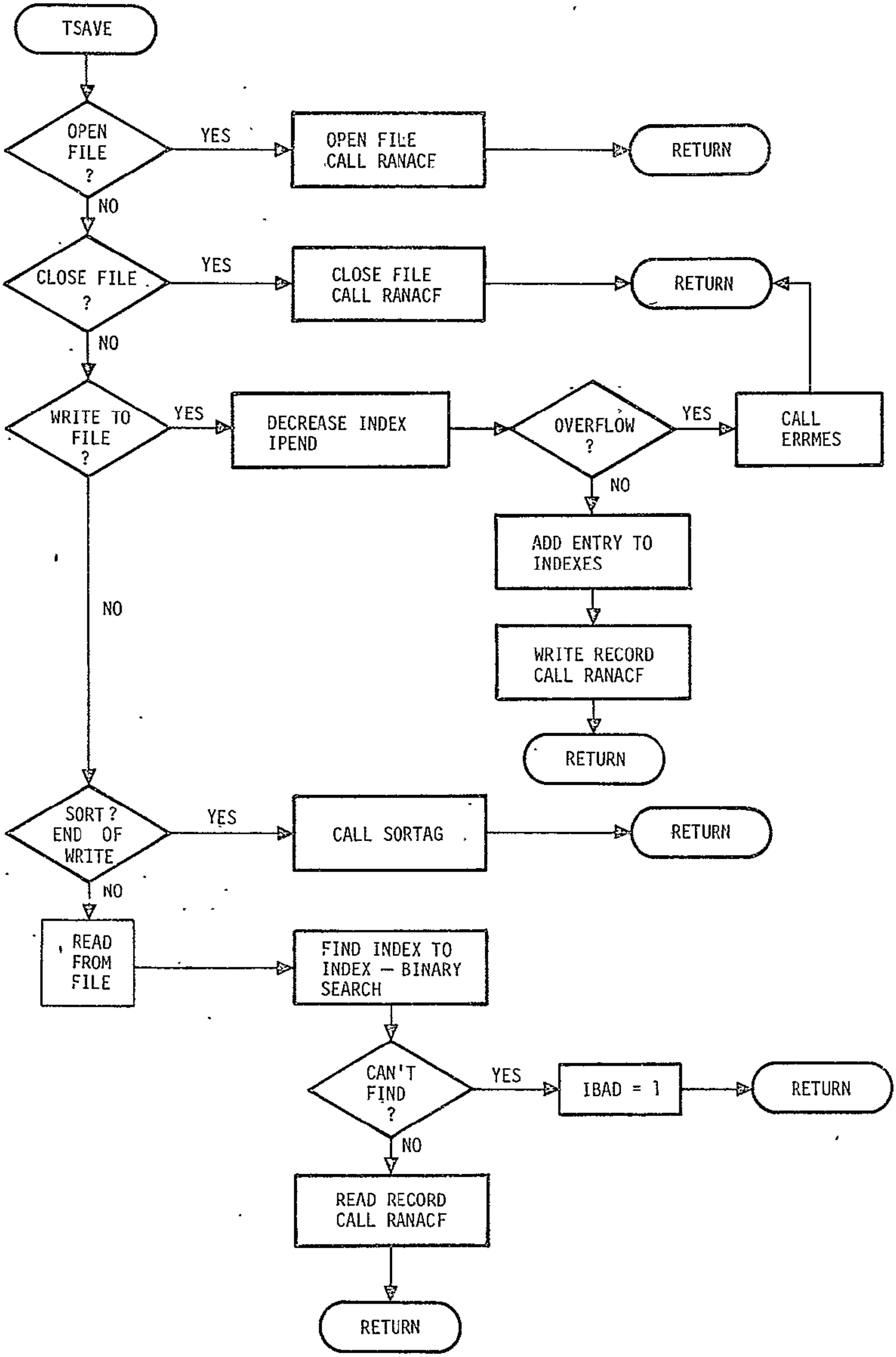
CALL ERRMES (4HCAMS, 5HTSAVE, 4, 1) to print error message.

CALL SORTAG (IPOINT, 1, IPEND, IPNT2) to sort files IPOINT, IPNT2 as IPOINT.

Local Variable Description:

IH	for binary search - the high limit
IL	for binary search - the low limit
ILOOK	for binary search - the current guess
L	the total no. of records allowed in file TACQ
NARG	no. of arguments for ERRMES routine (in COMMON /ARGLST/)

SUBROUTINE TSAVE BLOCK DIAGRAM



SUBROUTINE CLASS

Purpose:

This subroutine adds the input classification error to the crop calendar error using the bias and sigma from the CAMERR file. It then computes the total classification error, getting a random number from a beta distribution. It also computes and saves values for the error report, if needed.

Input:

/CAMERR/	PW(3, 4) BERR(3, 4) SIGERR(3, 4)	from CAMERR input file: bias and sigma for input classification error
/CNTRL/	PRINTF	print flag
/CAMSCM/	IREP	print flag for error report

See also linkage.

Output:

/ERROR/	CLTOT(TYPE) CLBIAS(TYPE) CLRAND(TYPE) ERTOT(TYPE) ERBIAS(TYPE) ERRAND(TYPE)	for error report: equations B. 1d-1g, 2d-2f from CAMS Problem Description
/TRAINS/	TV(TYPE)	

See also linkage.

Linkage:

CALL CLASS (SEED(2), TYPE, WINDOW, M, BCC, SIGCC, XI)

Inputs:

SEED(2)	double precision random no. seed used to generate classification error random no. from beta distribution
TYPE	integer, which component of mixed crops (1 = wheat, 2 = mixed, 3 = other; for model 2, 1 = wheat only used)
WINDOW	integer, which window current acquisition in (1, 2, 3 or 4)
M	output from MULTI subroutine, multi-temporal error
BCC SIGCC	outputs from CROP subroutine, crop calendar error

Outputs:

XI	total error factor
SEED(2)	double precision random no. seed after use

If error in BETA routine, prints message:

BETA DISTRIBUTION ERROR - FLAG = X

Subroutines Used:

CALL BETAD (SEED(2), XBAR, SIGMA, XI, 0, IER) to get random number from beta distribution.

SQRT(X) square root function

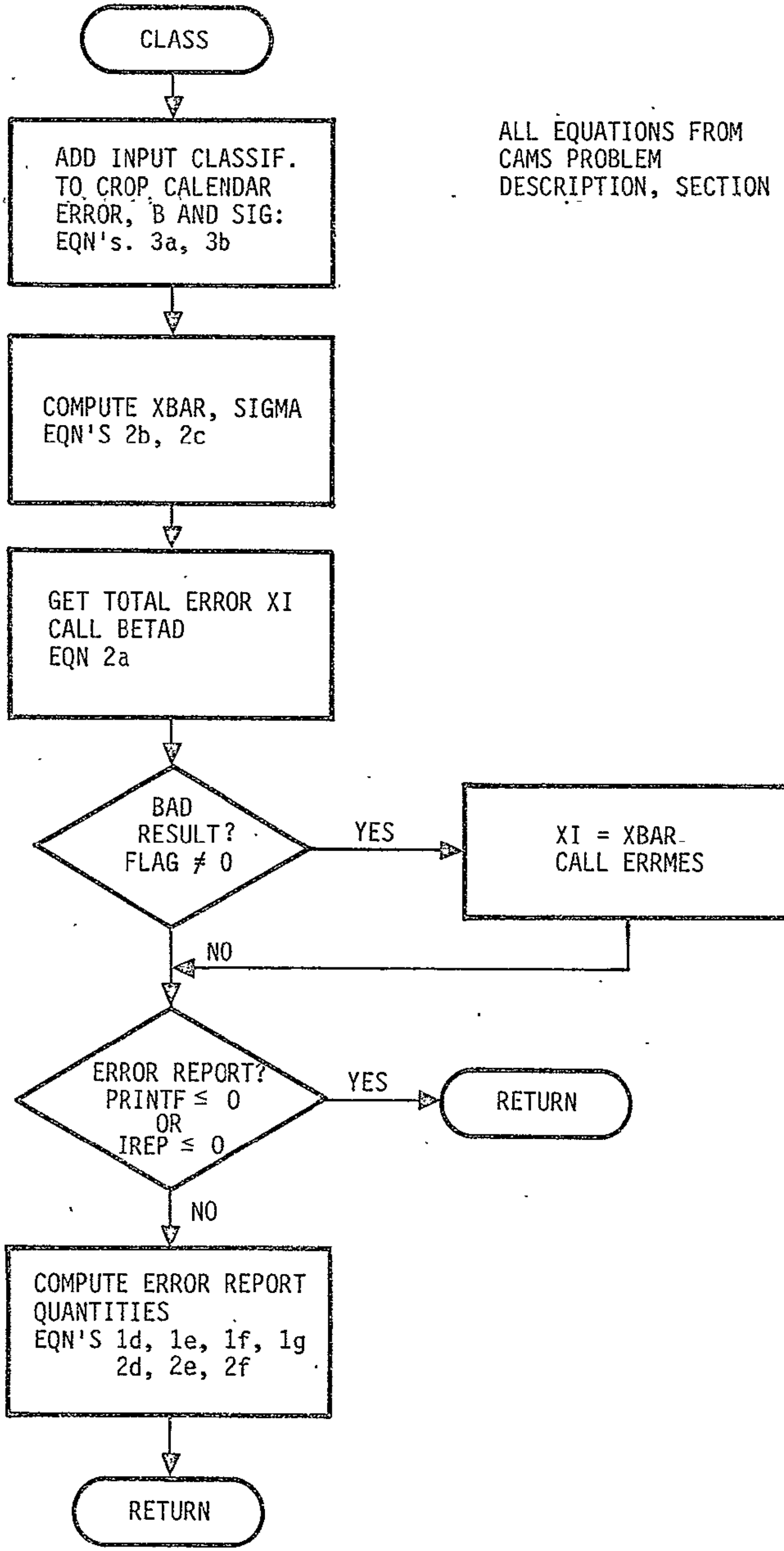
CALL ERRMES (4HCAMS, 5HCLASS, 4, 0) to report error in BETA routine.

Local Variable Description:

S, SIG	bias and sigma of input classification plus crop calendar errors, equations 3a, 3b
XBAR, SIGMA	total error bias and sigma, equations 2b, 2c
IER	

SUBROUTINE CLASS BLOCK DIAGRAM

ALL EQUATIONS FROM
CAMS PROBLEM
DESCRIPTION, SECTION 3.3, B



SUBROUTINE INPT

Purpose:

This subroutine sets up the next records to process on the five input files. It keys on the SEGTRU file. The data is set up differently depending on the model being used for some files. Some range checks are done. The reading of each file can be bypassed to handle error condition bypassing and the data requirements of each pass (training segments, ordinary segments, special cases - no acquisition file or no training segment ID match). As a side effect of the ordinary segment read pass, training segments are outputted to the output file CAMSF. At the return of INPT to CAMS, all necessary data is present and ready to use.

Input:

/FILES/	TACQ SEGTRU ACQUIS CAMERR CROPW SIGEXT CAMSF	logical file unit nos., integers
	LCAMSF	length of output record
/INDX/	INDEX	index for RA scratch file TACQ
/LEMCM/	ENDZ ENDR	zone and region to end processing at, integers
/CAMSCM/	IWIND	which window to use for recalculation of PT(M) if needed
	IMODEL	which model (1 or 2)

See also linkage.

Output:

Records for each input file are outputted, unless bypassed, in COMMON blocks. See Section 2.4 of the Users Manual for the file descriptions, and Programmers Manual for the COMMON block descriptions.

/SEGTRU/	COUN4	
	IREG4	
	IZONE4	holds 1 record from SEGTRU file
	ISTR4	
	ISUB4	
	ISEG4	
	IT	
	IPRIOR(6)	
	ISPW	
	PT(2)	PT(2) will be recalculated if file input out of range

If $PT(2) * PW(2, IWIND) < (PT(2) + PT(1) - 100)$

then $$PT(2) = \frac{100 - PT(1)}{100 - PW(2, IWIND)}$$

If $PT(2) * PW(2, IWIND) > PT(1)$ then

$$PT(2) = \frac{PT(1)}{PW(2, IWIND)}$$

/CAMERR/	COUN2 IREG2 IZONE2 ISTRA2 ISUBST2 ISEG2 PW(3, 4) BERR(3, 4) SIGERR(3, 4)	holds 1 record from CAMERR file if model 1, last 2 words of file ignored (B_{PW}, σ_{PW}) if model 2, middle 6 words ignored from file ($B_{W/W}$, $\sigma_{W/W}$, $B_{W/O}$, $\sigma_{W/O}$, $B_{W/M}$, $\sigma_{W/M}$) and B_{PW}, σ_{PW} stored in wheat dimension of BERR, SIGERR; also PW for wheat reset to PT (wheat)/100; PW for mixed reset to 0
/ACQUIS/	COUN1	
	IREG1	
	IZONE1	holds 1 record from ACQUIS file
	ISTRA1	
	ISUBST1	
	ISEG1	
	IWIN(4, 25)	
	ITOTAL	
/CROPW/	COUN3	
	IREG3	
	IZONE3	
	ISTRA3	
	ISUBST3	
	START(2, 4) END(2, 4) SD(2) ERR(2, 5)	read in groups; see file description, Section 2.4 of the Users Manual.

/SIGEXT/	COUN5 IREG5 IZONE5 ZB(3, 2) ZSIG(3, 2, 6)	holds 1 record from SIGEXT file if model 1, ignore model 2 data if model 2, store in wheat dimension and ignore model 1 data ZB(I, 1) = ZB(I, 1) + 1 for all ZB's for dimension 1
/ARGLST/	NERRS NFATAL NPERRS	error count passed back from ERRMLT subroutine
/INDX/	INDEX	index for RA scratch file TACQ
/STATS/	NREC(2) NREC(4) NREC(6) NREC(7) NCAMSR	no. of records processed for input files

See also linkage.

Linkage:

CALL INPT (ISEG, IACQU, ICAMER, ICROPW, ISGEX, IPASS, IDONE, IEND)

Inputs:

ISEG IACQU ICAMER ICROPW ISGEX	flags for input files 0 read file >0 skip reading file
IPASS	which pass, 0 = special cases 1 = training segment pass 2 = ordinary segment pass

Outputs:

IDONE	flag 0 = normal return 1 = end of file reached 2 = error return
IEND	flag 0 = end zone has not been reached 1 = end zone has been reached

Subroutines Used:

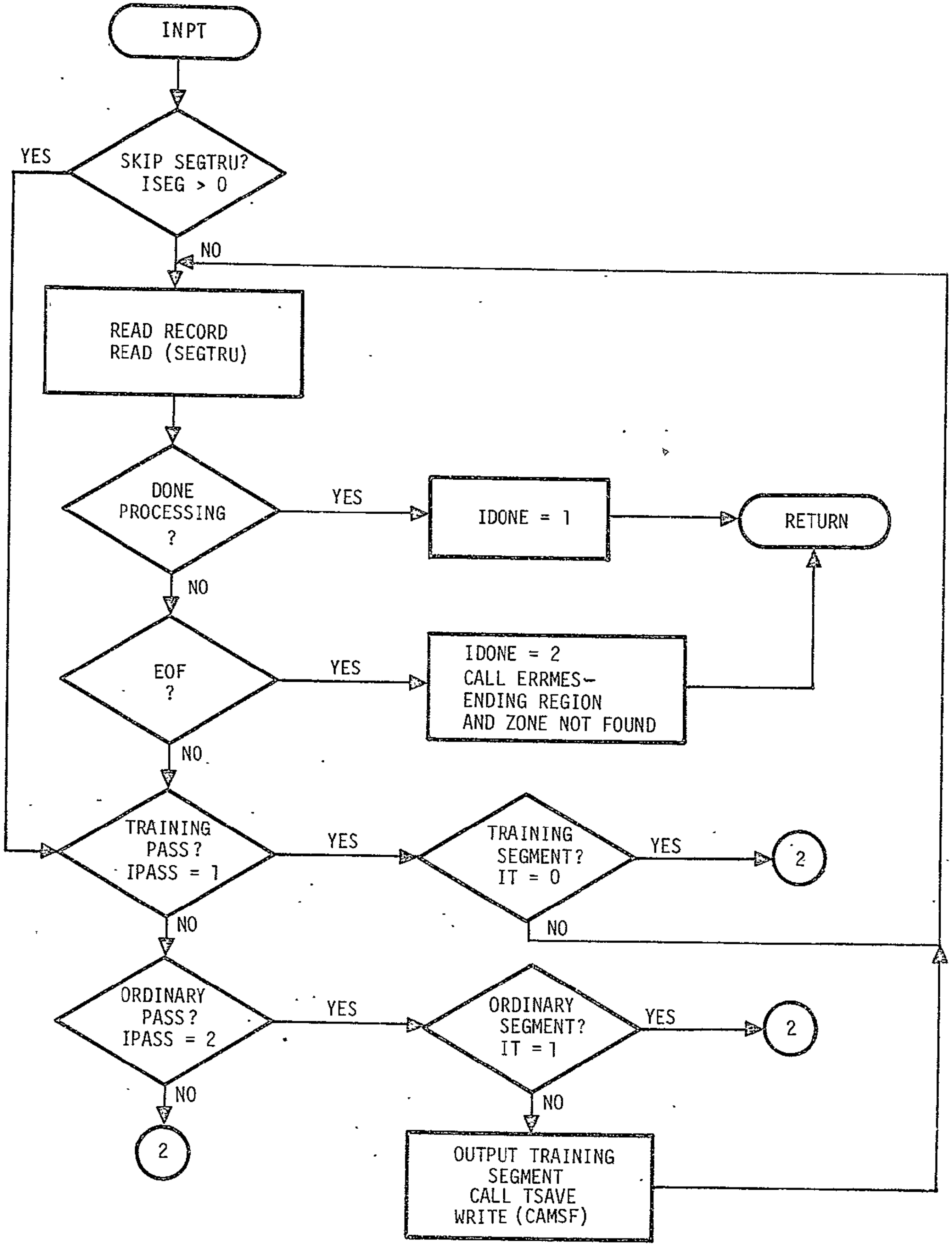
CALL ERRMES (4HCAMS, 3HINP, 8, 1) to report error message.

CALL TSAVE (ISEG, 1, IBAD) to write record from scratch RA file to output file.

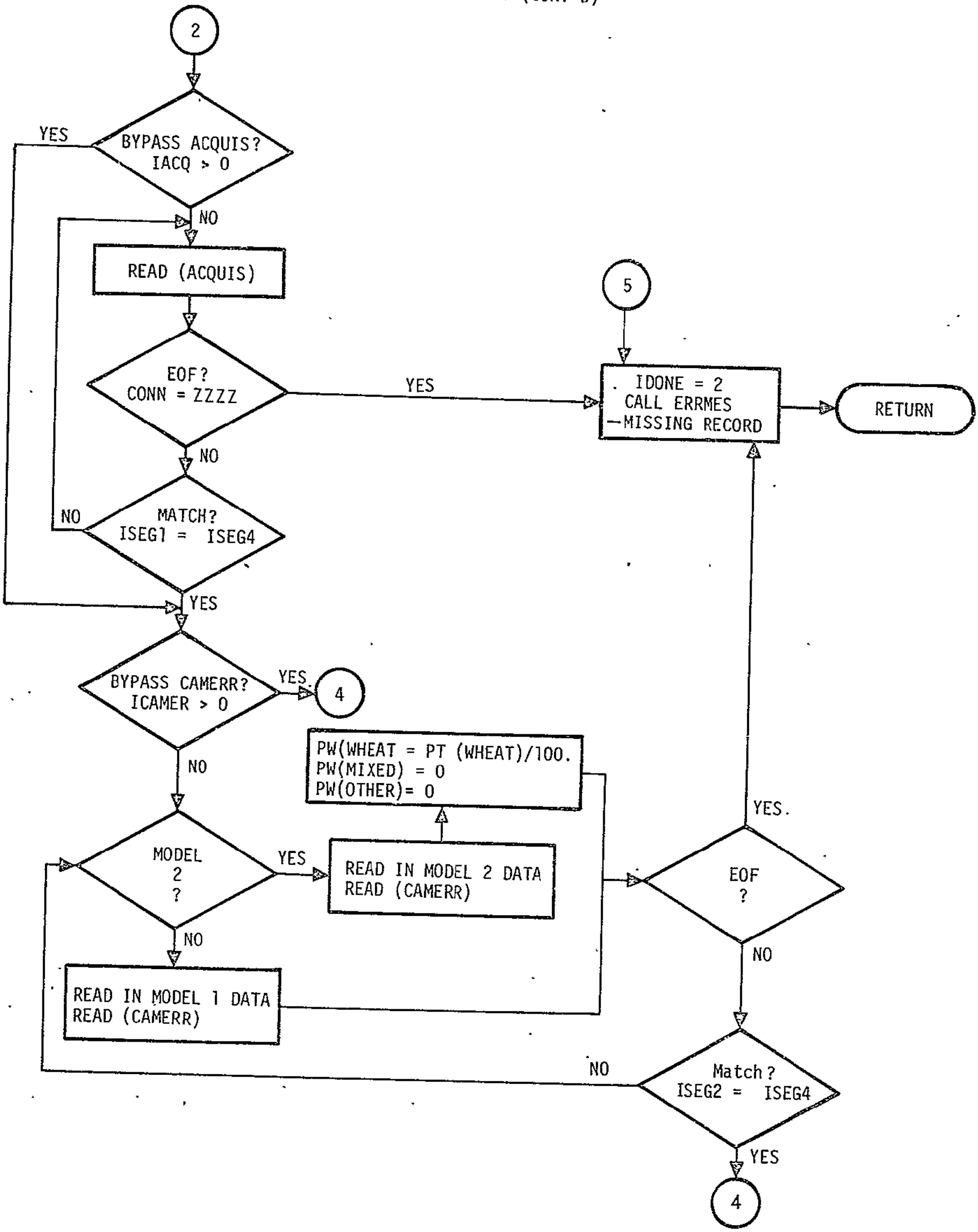
Local Variable Description:

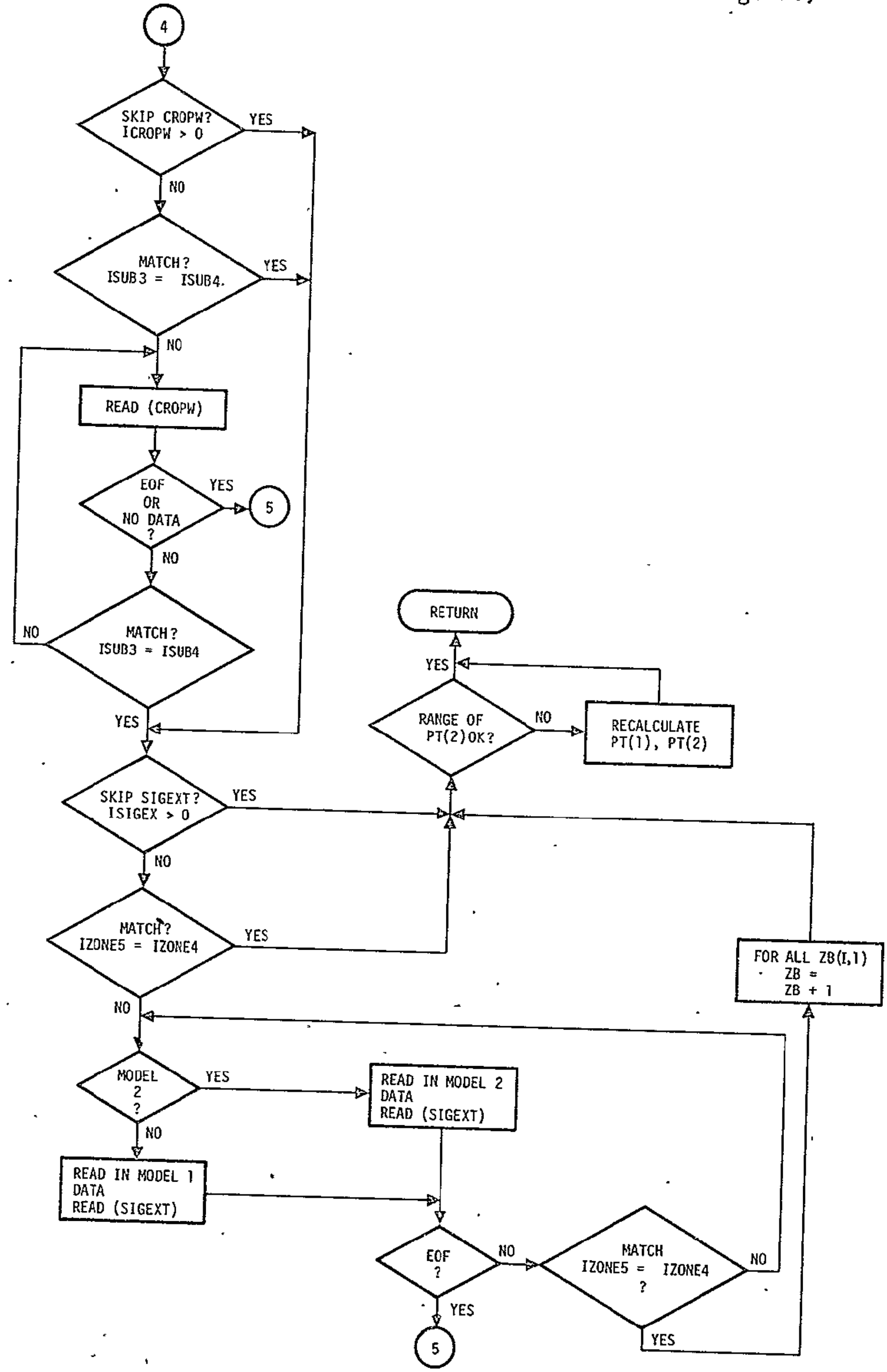
W	=1, wheat component
M	=2, mixed component
/ARGLST/NARG	for error subroutine ERRMES
ITOTAL	no. of filler words
ISKIP	used to skip over words
IEND	flag to signal end zone has been reached, 0 = not yet, 1 = has been reached

SUBROUTINE INPT BLOCK DIAGRAM



INPT (CONT'D)


2
BYPASS ACQUIS?
IACQ > 0
YES
NO
READ (ACQUIS)
EOF?
CONN = ZZZZ
YES
NO
MATCH?
ISEG1 = ISEG4
NO
YES
5
IDONE = 2
CALL ERRMES
—MISSING RECORD
RETURN
BYPASS CAMERR?
ICAMER > 0
YES
4
NO
PW(WHEAT = PT (WHEAT)/100.
PW(MIXED) = 0
PW(OTHER)= 0
MODEL
2
?
YES
READ IN MODEL 2 DATA
READ (CAMERR)
NO
READ IN MODEL 1 DATA
READ (CAMERR)
EOF
?
YES.
NO
Match?
ISEG2 = ISEG4
NO
YES
4


4
SKIP CROPW?
ICROPW > 0
YES
NO
MATCH?
ISUB3 = ISUB4.
YES
NO
READ (CROPW)
EOF
OR
NO DATA
?
YES
5
NO
MATCH?
ISUB3 = ISUB4
NO
YES
SKIP SIGEXT?
ISIGEX > 0
YES
NO
MATCH?
IZONE5 = IZONE4
YES
NO
MODEL
2
?
YES
NO
READ IN MODEL 2
DATA
READ (SIGEXT)
READ IN MODEL 1
DATA
READ (SIGEXT)
EOF
?
NO
YES
5
MATCH
IZONE5 = IZONE4
?
NO
YES
FOR ALL ZB(I,1)
ZB =
ZB + 1
RETURN
YES
RANGE OF
PT(2) OK?
NO
RECALCULATE
PT(1), PT(2)

CAS SUBROUTINE DESCRIPTIONS

Subroutine CAS

Purpose:

CAS is the driver for the CAS simulator. It controls the looping for the bio-windows and prediction dates, calling CASPP to perform the CAS computations for each bio-window and each prediction date.

Input:

Quantity	Common Block	Source
BWIND	CASCM	CASIN
NPDATE	CASCM	CASIN
PRDATE	CASCM	CASIN
PRINTF	CONTRL	LEM (SETPRF)
NFATAL	ARGLST	ERRMES

Output:

Quantity	Common Block	Source
IBW	CASFLG	GROUP, CAS2
IPD	CASFLG	GETYS, GROUP, CAS2
PPFLG	CASFLG	GETYS, GROUP CAS2
PPDATE	CASFLG	GETYS, GROUP

Linkage:

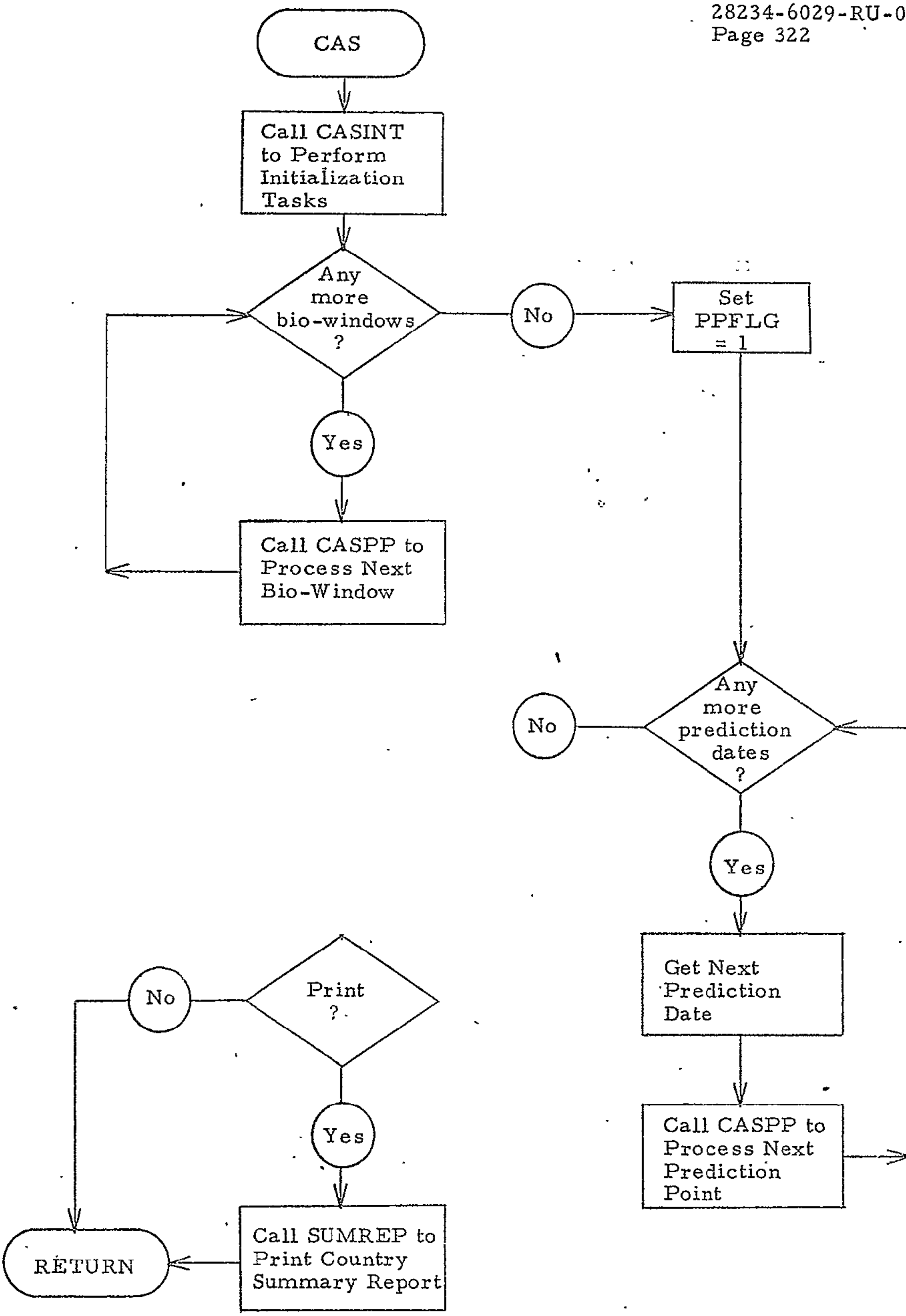
CALL CAS

Subroutines Used:

CASINT
CASPP
SUMREP

Local Variables:

None.



CAS Flow Diagram

Subroutine CASPP

Purpose:

Subroutine CASPP performs the first pass CAS computations generating data sets 1-9. It also calls CAS2 to compute the estimated group 1, 2 area variances and calls CAS3 to generate data sets 10-19.

Input:

Quantity	Common Block	File	Source
NFATAL	ARGLST		ERRMES
IRREG	CASFLG		CASINL
IPP	CASFLG		CASINT
LDS 4	CASFLG		Block Data
LDS 7	CASFLG		Block Data
LDS 8	CASFLG		Block Data
LDS 9	CASFLG		Block Data
NSTART	CNTRL		LEM
IMXSEG	CONST		Block Data
ENDFIL	CONST		Block Data
STRATA	DSET4	YESOUT	GETYS
ZONE	DSET7	YESOUT	GETYS
REGION	DSET8	YESOUT	GETYS
TWAR	DSET8		DS7
HWAR1	DSET8		DS7
EWAR1	DSET8		DS7
DSET8	DSET8		DS7
CASDSF	FILES		Block Data
LCASDS	FILES		Block Data
ISUBH2	FILES1		Block Data
LSUBH2	FILES1		Block Data
LIXSSH	IXSUBH		Block Data
LIXCDS	IXCDSF		Block Data
ENDR	LEMCM		INPUT
ENDZ	LEMCM		INPUT
SSHDTA	SSHDTA	ISUBH2	

Quantity	Common Block	File	Source
COUN2	SSHDTA	ISUBH2	See description of /SSHDTA/
IREG2	SSHDTA	ISUBH2	
⋮	⋮	⋮	
CV4	SSHDTA	ISUBH2	
CLASS	SSHDTA	ISUBH2	
NT ≡ITER	STATS	-	LEM
YSTR	YESDTA	YESOUT	GETYS

Output:

Quantity	Common Block	File	Used By
IPP	CASFLG		CAS2
KSUB	CASFLG		DS123
DSET4	DSET4		DS456, CAS2
DSET7	DSET7		DS7, CAS2
DSET8	DSET8		CAS2
DSET9	DSET9		CAS2
ARG(1)	ARGLST		ERRMES
NRSSH	CASFLG		WRAPUP
IRREG	CASFLG		RANACF
NREGS	CASFLG		CAS2
ER	DSET8		CAS2
EC	DSET9		CAS2
M1M2ZC	DSET9		CAS2
HWAC1	DSET9		CAS2
EWAC1	DSET9		CAS2

Linkage:

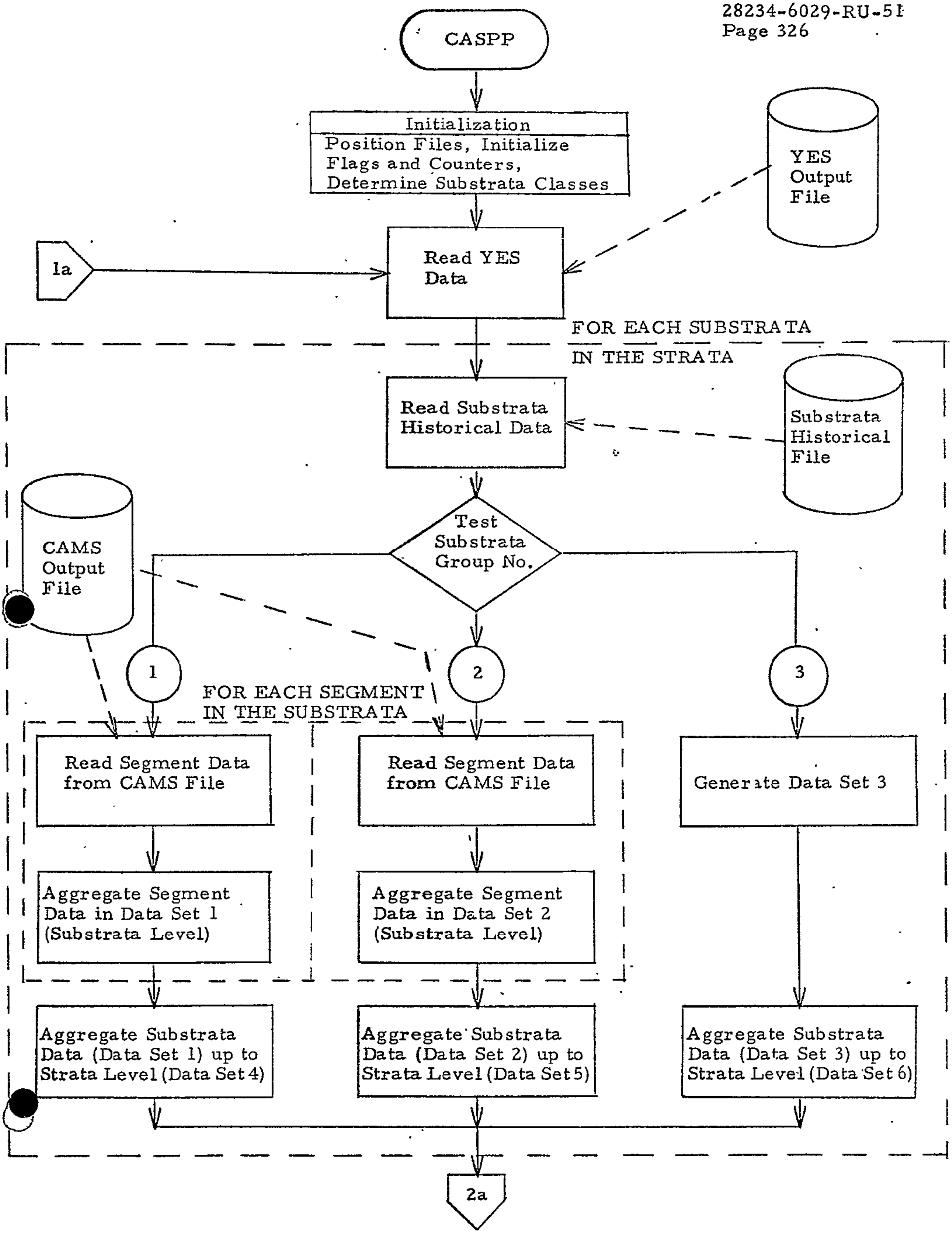
CALL CASPP

Subroutines Used:

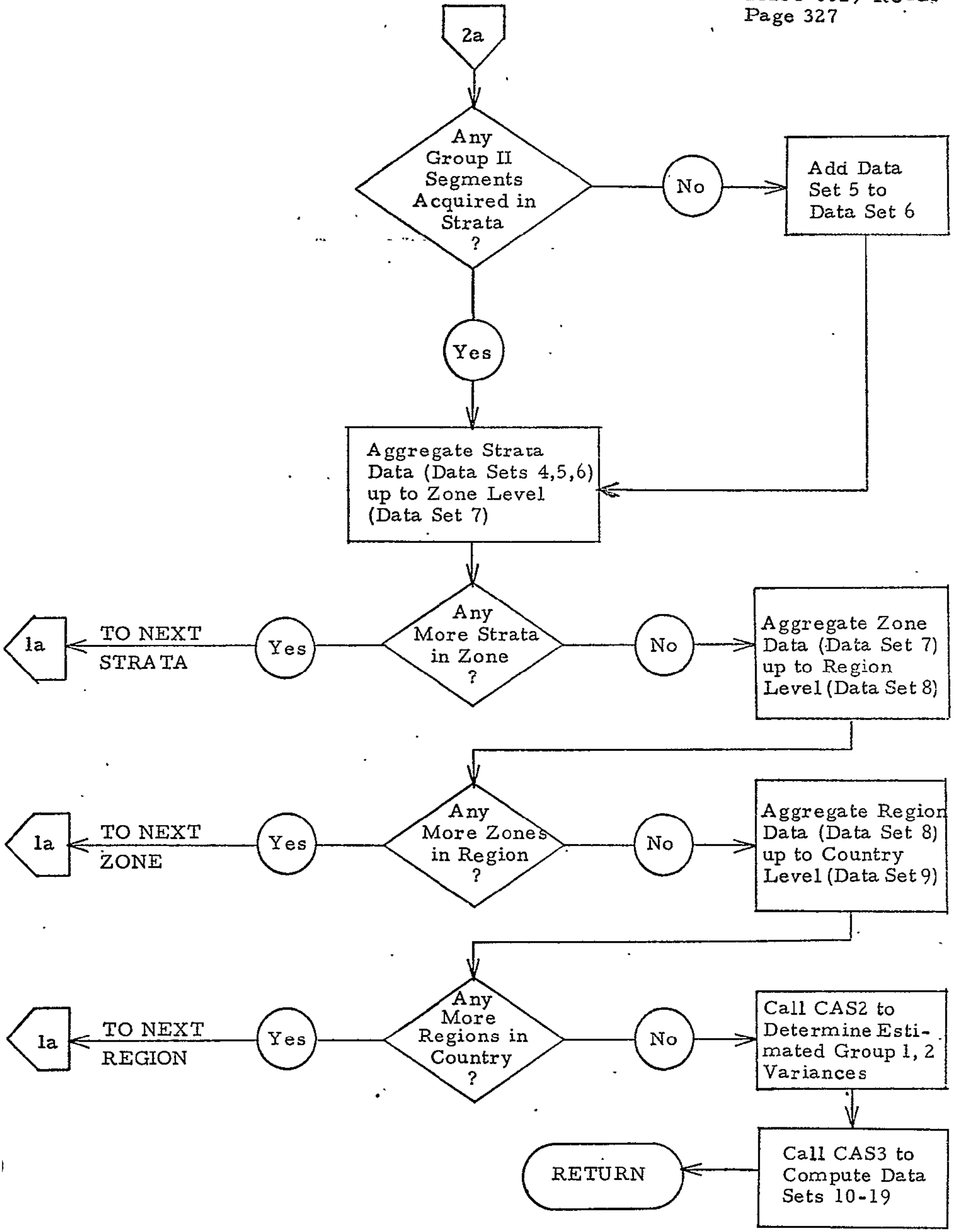
CASINL
CLASSN
GETYS
ERRMES
DS123
DS456
DS7
RANACF
CAS2
CAS3

Processing:

See Flow Diagram on the following pages.



CASPP Flow Diagram (Sheet 1 of 2



CASPP Flow Diagram (Sheet 2 of 2)

Subroutine CASOUT

Purpose:

Subroutine CASOUT processes the CAS Output data for the CAS Area and Production Summary Report and the CAS Country Summary Report. It converts data to output units, computes mean values, and on option prints a portion of the Area and Production Summary Report.

Input:

Quantity	Common Block	Source
ILEVEL	Calling sequence parameter	CAS2, DS10
OUTP	FILES	Block Data
AREACF	CASCM	CASIN
YCF	CASCM	CASIN
PRDCF	CASCM	CASIN
APREP	CASCM	CASIN
CASCUM	CASCUM	DS10, CAS2
IREG		
IZONE		
ISTRA		
HWA		
TWA		
EWA		
AERR		
AVAR		
TPROD		
EPROD		
PRERR		
PRVAR		
TY		
EY		

Quantity	Common Block	Source
YERR	CASCUM	DS10, CAS2
ANAV		
ANPRV		
SQAER		
SQPER		
SQYER		
CLEWA		
CLEPRD		
CLATEC		
CLPTEC		
CLATWC		
CLPTWC		
IPP	CASFLG	LEM
ENDREG	CASFLG	CAS2
NLINE	PAGECM	PAGER
MXLINE	PAGECM	PAGER
NT	STATS	LEM
NSTRAZ	DSET7	CAS2

Output:

1) Printed Output

Quantity	Common Block
IREG	CASCUM
IZONE	CASCUM
ISTRA	CASCUM
TWA	CASCUM
EWA	CASCUM
CT1	
CT2	
CT3	
M1J	
M2J	

Quantity	Common Block
CVAEPT	
CVEPTA	
TY	CASCUM
EY	CASCUM
SDPER	CASCUM
TPROD	CASCUM
EPROD	CASCUM
CVPEPT	
CVEPTP	

2) Output to Common Blocks

Quantity	Common Block	Used By
SQAER	CASCUM	DS18
SQPER	CASCUM	DS18
TWA	CASCUM	DS18
EWA	CASCUM	DS18
AERR	CASCUM	DS18
TPROD	CASCUM	DS18
EPROD	CASCUM	DS18
PRERR	CASCUM	DS18
CSUMR	SUMDTA	SUMREP

Note: The quantities SQAER, SQPER, ... , PRERR are converted from the internal units (hectares and quintals) to the appropriate output units. In addition, the mean values of TWA, EWA, AERR, TPROD, EPROD, and PRERR are computed.

Linkage:

CALL CASOUT (ILEVEL)

where

ILEVEL	=0	for country,
	=-1	for region,
	=-2	for zone,
	=+n	for n^{th} strata in zone

Subroutines Used:

APHDR - Prints Area and Production Summary Report headers

PAGER - Automatic paging routine

SQRT - Square root routine

Processing:

1. Convert units from internal units to appropriate output units and compute mean values; e.g.,

   HWA = HWA * AREACF/NT
   ⋮
   AVAR = AVAR * AREACF**2/NT

   TPROD = TPROD * PRDCF/NT
   ⋮
   PRVAR = PRVAR * PRDCF**2/NT

   TY = TY * YCF/NT
   ⋮
   M1J = M1J/NT
   ⋮
   CT3 = CT3/NT

   ANAV = ANAV * AREACF**2/NT

   ANPRV = ANPRV * PRDCF**2/NT

   SQAER = SQAER * AREACF**2

   SQPER = SQPER * PRDCF**2

   SQYER = SQYER * YCF**2

2. If ILEVEL = 0 (country level), compute average values of confidence levels; e.g.,

   CLEWA = CLEWA/NT

3. Compute coefficients of variance for Country Summary Report

   - CVAEPT = $\frac{\sqrt{AVAR}}{TWA} * 100$
     [CV Area Est. (% True)]

   - CVPEPT = $\frac{\sqrt{PRVAR}}{TPROD} * 100$
     [CV Prod. Est. (% True)]

- $$CVEPTA = \frac{\sqrt{\dfrac{SQAER - AERR^2 * NT}{NT - 1}} * 100}{TWA}$$

$\begin{bmatrix}\text{Area} \\ \text{CV Error} \\ \text{(\% True}\end{bmatrix}$

- $$SDPEP = \sqrt{\frac{SQYER - YERR^2 * NT}{NT - 1}}$$

$\begin{bmatrix}\text{Yield} \\ \text{Std. Dev.} \\ \text{(\% True)}\end{bmatrix}$

- $$CVEPTP = \frac{\sqrt{\dfrac{SQPER - PRERR^2 * NT}{NT - 1}} * 100}{TPROD}$$

$\begin{bmatrix}\text{Prod.} \\ \text{CV Error} \\ \text{(\% True)}\end{bmatrix}$

4. If APREP $\neq$ 0, then one data line is printed for the Area and Production Summary Report. For the first strata of each zone, the data line may be preceded by several lines of identification information such as the input problem header, the bio-window number or prediction date, the current iteration number, and data levels.

5. Finally, if ILEVEL = 0 (country level), values are saved in the array CSUMR for the current prediction for later printout in subroutine SUMREP.

Subroutine CAS2

Purpose:

Subroutine CAS2 computes the area variances and analytic area variances for all strata which have acquired segments. CAS2 also aggregates quantities at the zone, region, and country levels which will be used to compute the area variances for strata without acquired segments.

Input:

Quantity	Common Block	File	Source
IPRD	CASCM		CASIN
PPFLG	CASFLG		CAS, CASINT
IBW	CASFLG		CAS
IPD	CASFLG		CAS
IPP	CASFLG		CAS
NREGS	CASFLG		CASPP
ISUBH2	FILES1		Block Data
LSUBH2	FILES1		Block Data
MXCLSS	FILES1		Block Data
DSET4	DSET4	ISUBH2	DS456
HWAS1	↓	↓	↓
XM1JS	↓	↓	↓
XCT1S	↓	↓	↓
ANVS1	↓	↓	↓
HWAS2	↓	↓	↓
XM2JS	↓	↓	↓
XCT2S	↓	↓	↓
ANVS2	↓	↓	↓
T	↓	↓	↓
HWAS3	↓	↓	↓
XCT3S	DSET4	ISUBH2	DS456

Input: (cont'd)

Quantity	Common Block	File	Source
DSET7	DSET7	ISUBH2	DS7
MIK2KZ	↓	↓	↓
NSTRAZ	↓	↓	↓
MIK2CL	↓	↓	↓
SSQ	DSET7	ISUBH2	DS7
DSET8	DSET8	ISUBH2	CASPP
MIK2KR	DSET8	ISUBH2	CASPP
NZONES	DSET8	ISUBH2	CASPP
DSET9	DSET9	ISUBH2	CASPP
MIK2KC	DSET9	ISUBH2	CASPP
QUTP	FILES		Block Data
CASDSF	FILES		Block Data
LCASDS	FILES		Block Data
LIXCDS	IXCDSF		Block Data
LIXSSH	IXSUBH		Block Data
SSHDTA	SSHDTA	ISUBH2	DS123
GRPNO	SSHDTA	ISUBH2	DS123
VMULTK	SSHDTA	ISUBH2	DS123
CLASS	SSHDTA	ISUBH2	CLASSN

Output:

Quantity	Common Block	File	Used By
NRSSH	CASFLG		RANACF
IRSTR	↓		↓
IRZONE	↓		↓
IRREG	CASFLG		RANACF
DSET4	DSET4	CASDSF	CAS3, DS10
V1V2S	↓	↓	↓ ↓
VARS	↓	↓	↓ ↓
ANVARS	DSET4	CASDSF	CAS3, DS10

Output: (cont'd)

Quantity	Common Block	File	Used By
DSET7	DSET7	CASDSF	CAS3, DS10
ANALVZ	↓	↓	↓ ↓
HWAZ3	↓	↓	↓ ↓
ESTVZ	DSET7	CASDSF	CAS3, DS10
DSET8	DSET8	CASDSF	CAS3, DS10
ANALVR	DSET8	CASDSF	CAS3, DS10
ESTVR	DSET8	CASDSF	CAS3, DS10
DSET9	DSET9	CASDSF	CAS3, DS10
ANALVC	DSET9	CASDSF	CAS3, DS10
ESTVC	DSET9	CASDSF	CAS3, DS10

Linkage:

CALL CAS2

Subroutines Used:

PAGER

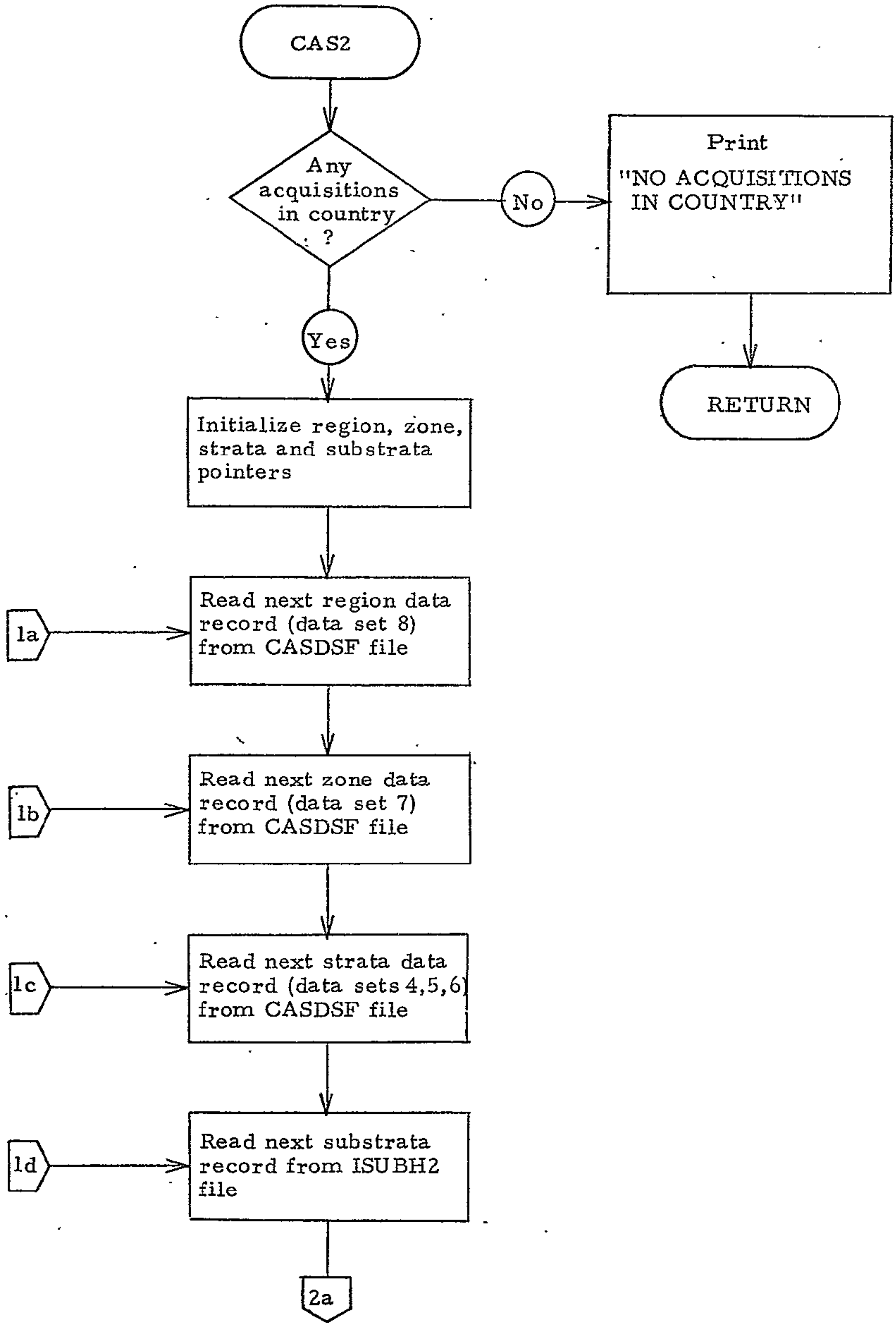
RANACF

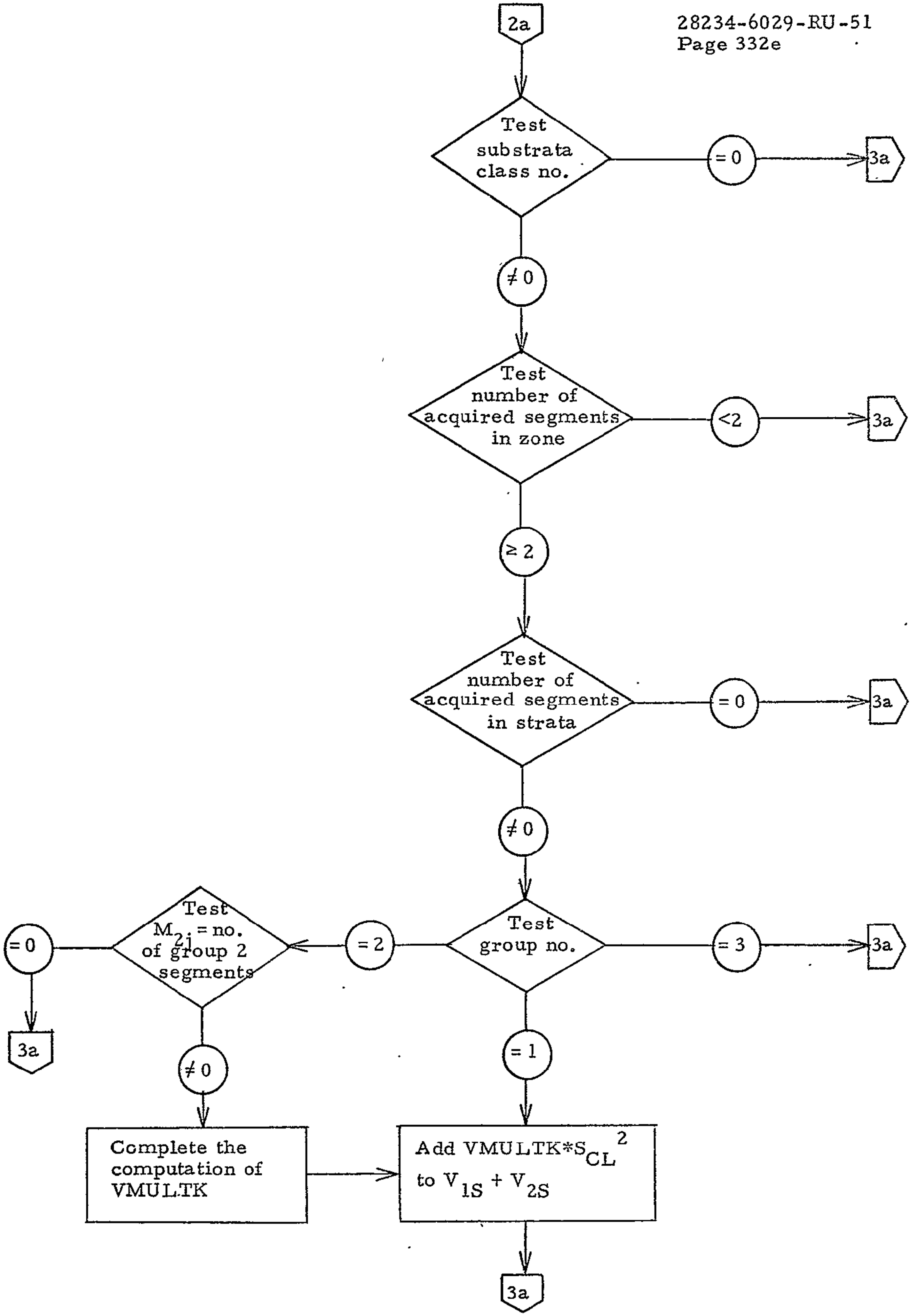
Local Variables:

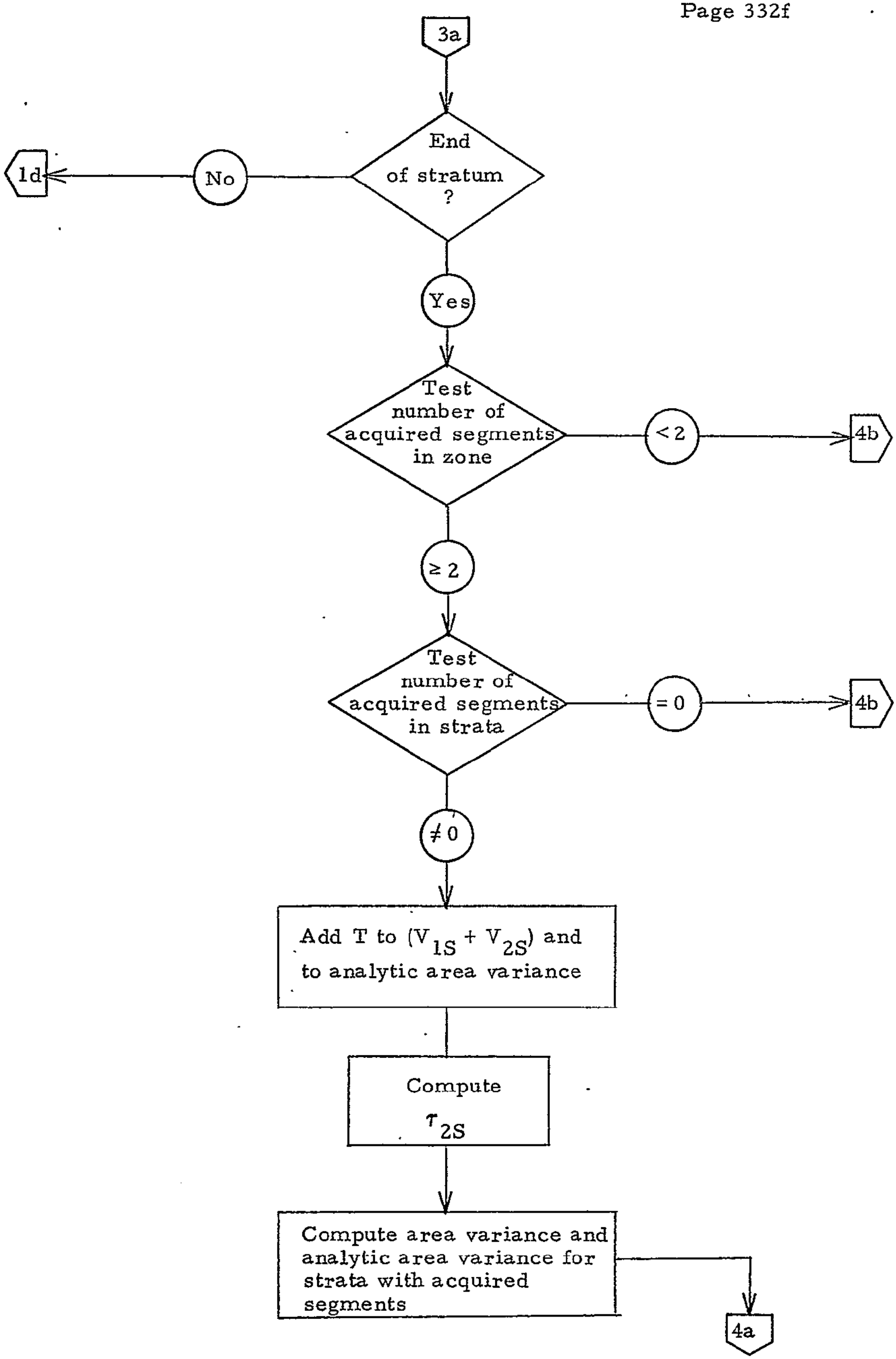
I	-	Index in DO loops
IREG	-	Region index
IZONE	-	Zone index
ISTRAZ	-	Strata index
NSUB	-	Number of substrata in the current stratum
ISUB	-	Substrata index
TAU2S	-	τ_S^2
ICL	-	Substrata class index

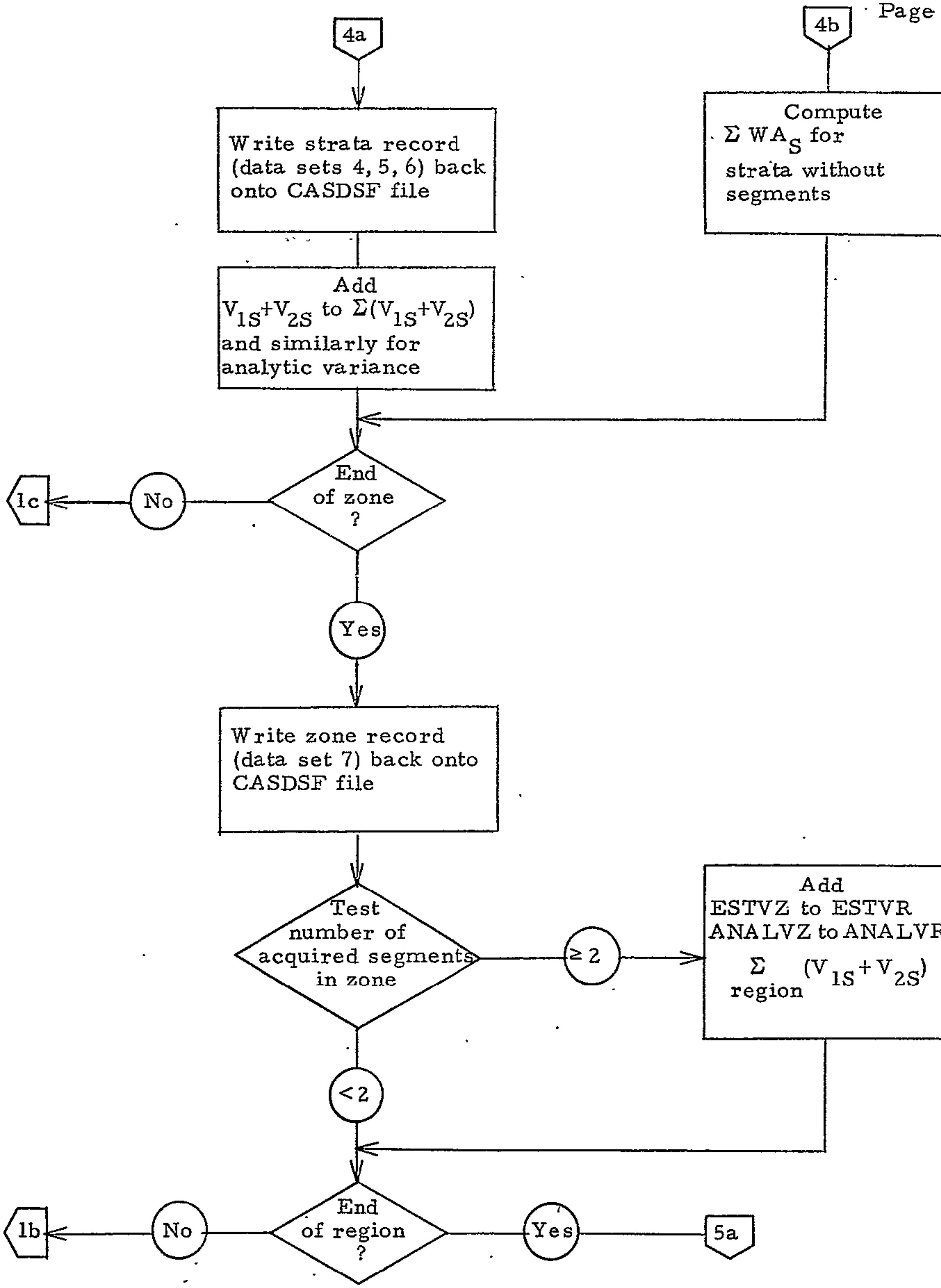
Processing:

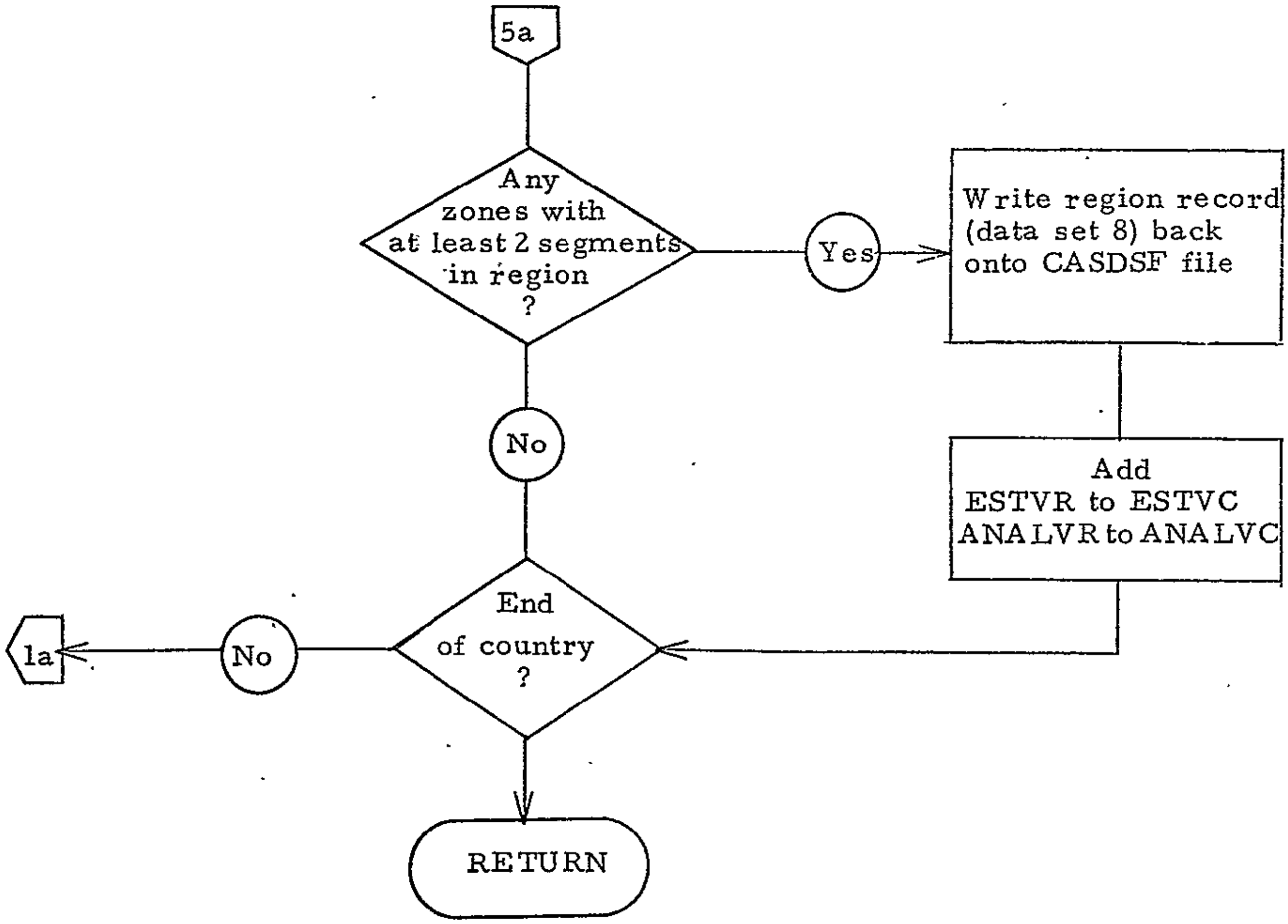
See Flow Diagram.


CAS2
Any acquisitions in country ?
No
Print "NO ACQUISITIONS IN COUNTRY"
RETURN
Yes
Initialize region, zone, strata and substrata pointers
1a
Read next region data record (data set 8) from CASDSF file
1b
Read next zone data record (data set 7) from CASDSF file
1c
Read next strata data record (data sets 4,5,6) from CASDSF file
1d
Read next substrata record from ISUBH2 file
2a


2a
Test substrata class no.
= 0
3a
≠ 0
Test number of acquired segments in zone
<2
3a
≥ 2
Test number of acquired segments in strata
= 0
3a
≠ 0
Test group no.
= 3
3a
= 2
Test M_{2j} = no. of group 2 segments
= 0
3a
≠ 0
Complete the computation of VMULTK
= 1
Add VMULTK*S_{CL}^2 to $V_{1S} + V_{2S}$
3a


3a
End of stratum ?
No
1d
Yes
Test number of acquired segments in zone
< 2
4b
≥ 2
Test number of acquired segments in strata
= 0
4b
≠ 0
Add T to (V_{1S} + V_{2S}) and to analytic area variance
Compute τ_{2S}
Compute area variance and analytic area variance for strata with acquired segments
4a


4a
4b
Write strata record (data sets 4, 5, 6) back onto CASDSF file
Compute Σ WA_S for strata without segments
Add V_1S+V_2S to Σ(V_1S+V_2S) and similarly for analytic variance
End of zone ?
No
1c
Yes
Write zone record (data set 7) back onto CASDSF file
Test number of acquired segments in zone
≥ 2
Add ESTVZ to ESTVR ANALVZ to ANALVR Σ region (V_1S + V_2S)
< 2
End of region ?
No
1b
Yes
5a


5a
Any zones with at least 2 segments in region ?
Yes
Write region record (data set 8) back onto CASDSF file
No
Add ESTVR to ESTVC ANALVR to ANALVC
End of country ?
No
1a
RETURN

Subroutine CAS3

Purpose:

CAS3 performs the final pass CAS computations generating data sets 10-19.

Input:

Quantity	Common Block	File	Source
NFATAL	ARGLST		ERRMES
DISTFF	CASCM		CASIN
APREP	CASCM		CASIN
IPRD	CASCM		CASIN
CASCUM	CASCUM	CASCUM	RWCASF
ICASC	CASCUM	CASCUM	RWCASF
DSET15	CASCUM	CASCUM	RWCASF
DSET16	CASCUM	CASCUM	RWCASF
DSET17	CASCUM	CASCUM	RWCASF
SQAERS	CASCUM	CASCUM	RWCASF
SQPERS	CASCUM	CASCUM	RWCASF
SQYERS	CASCUM	CASCUM	RWCASF
⋮	⋮	⋮	⋮
SQYERC	CASCUM	CASCUM	RWCASF
PPFLG	CASFLG		CAS
IBW	CASFLG		CAS
IPD	CASFLG		CAS
IPP	CASFLG		CAS
NREGS	CASFLG		CAS
LDS 11	CASFLG		CAS
LDS 12	CASFLG		CAS
LDS 13	CASFLG		CAS
LDS 15	CASFLG		CAS
LDS 16	CASFLG		CAS
LDS 17	CASFLG		CAS

Quantity	Common Block	File	Source
PRINTF	CNTRL		SETPRF
DSET7	DSET7	CASDSF	CASPP
ZONE	DSET7	CASDSF	CASPP
NSTRAZ	DSET7	CASDSF	CASPP
DSET8	DSET8	CASDSF	CASPP
REGION	DSET8	CASDSF	CASPP
DSET9	DSET9	CASDSF	CASPP
DSET11	DSET11		DS10
TWAZ	DSET11		DS10
EWAZ	DSET11		DS10
TPRODZ	DSET11		DS10
EPRODZ	DSET11		DS10
OUTP	FILES		Block Data
CASDSF	FILES		Block Data
LCASDS	FILES		Block Data
LIXCDS	IXCDSF		Block Data
NTRIAL	LEMCM		LEMCF
MXLINE	PAGECM		Block Data
NT≡ITER	STATS		LEM

Output:

Quantity	Common Block	File	Used By
CASCUM	CASCUM	CASCUM	RWCASF
DSET15	CASCUM	CASCUM	RWCASF
DSET16	CASCUM	CASCUM	RWCASF
DSET17	CASCUM	CASCUM	RWCASF
IRSTR	CASFLG		DS10
IRZONE	CASFLG		RWCASF, RANACF
IRREG	CASFLG		RWCASF, RANACF
ENDREG	CASFLG		CASOUT
DSET11	DSET11	CASDIS	RWDISF
AERRZ	DSET11	CASDIS	RWDISF
PRERRZ	DSET11	CASDIS	RWDISF

Quantity	Common Block	File	Used By
TYZ	DSET11	CASDIS	RWDISF
EYZ	DSET11	CASDIS	RWDISF
YERRZ	DSET11	CASDIS	RWDISF
DSET12	DSET12	CASDIS	RWDISF
AERRR	DSET12	CASDIS	RWDISF
PRERRR	DSET12	CASDIS	RWDISF
PRVARR	DSET12	CASDIS	RWDISF
TYR	DSET12	CASDIS	RWDISF
EYR	DSET12	CASDIS	RWDISF
YERRR	DSET12	CASDIS	RWDISF
DSET13	DSET13	CASDIS	RWDISF
NPAGE	PAGECM		PAGER

Linkage:

CALL CAS3

Subroutines Used:

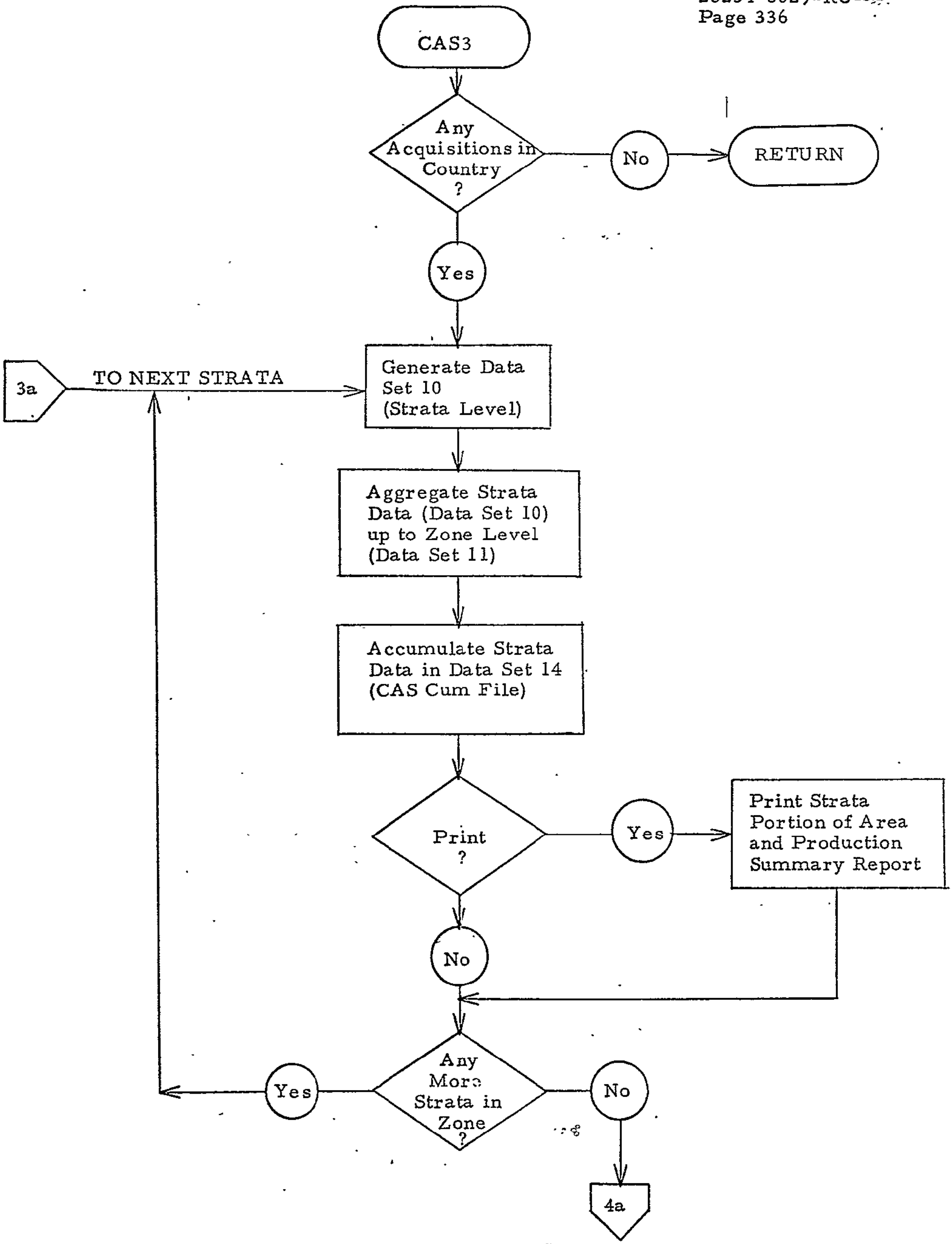
PAGER
RANACF
DS10
RWCASF
RWDISF
CASOUT
CONFL
DS10

Local Variables:

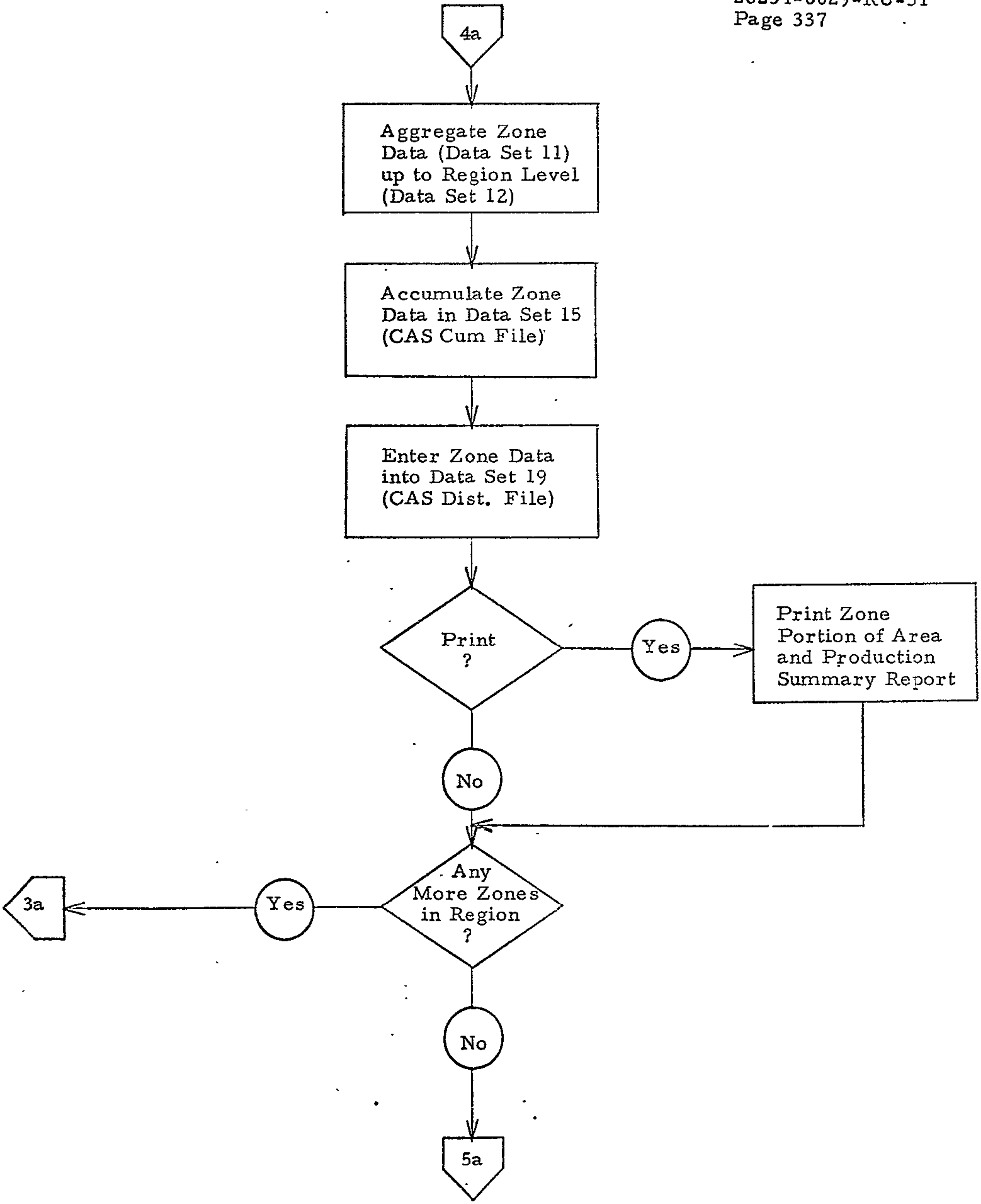
I - Index in DO loops
IREG - Region counter
IZONE - Zone counter (within a region)

Processing:

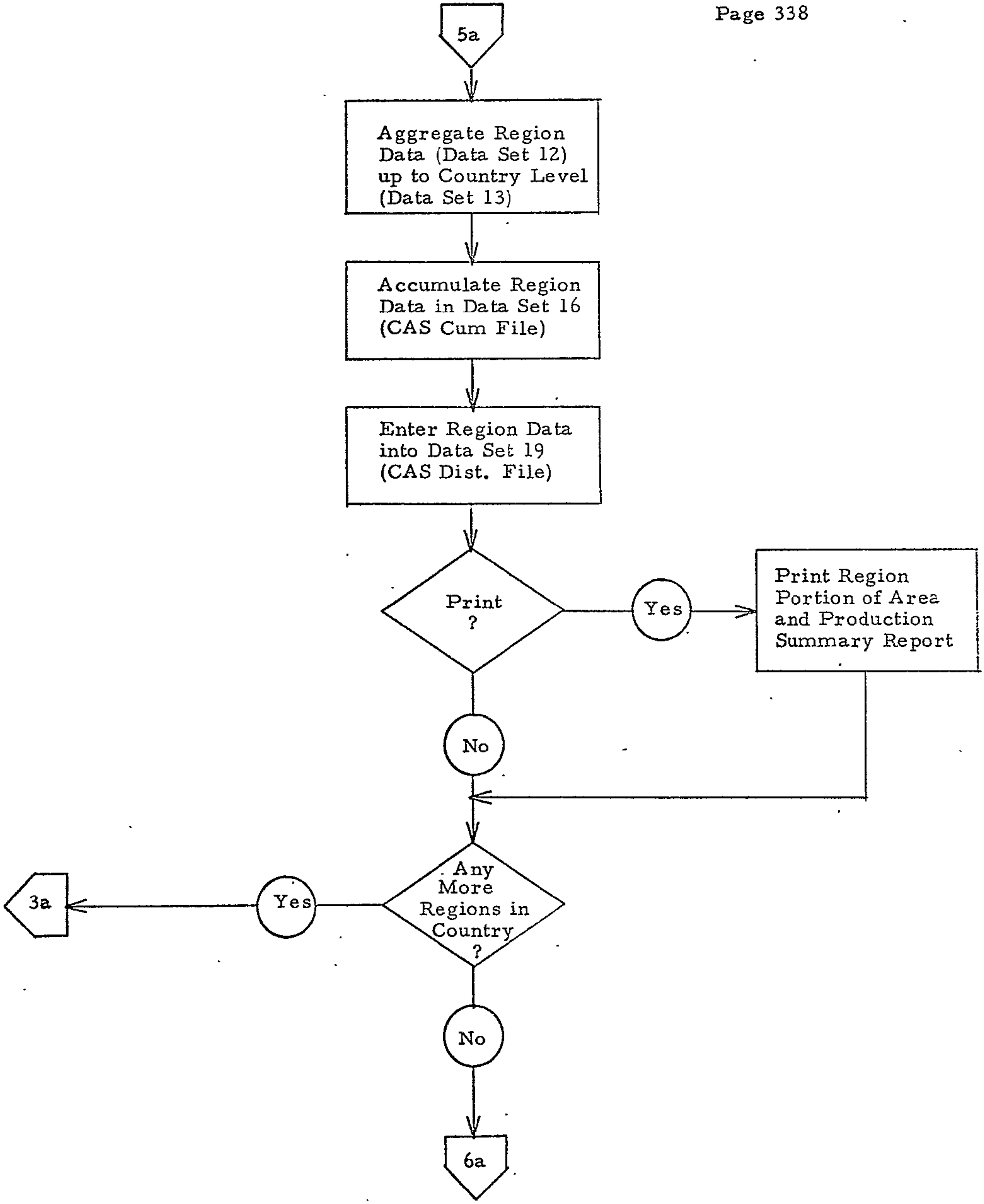
See Flow Diagram.



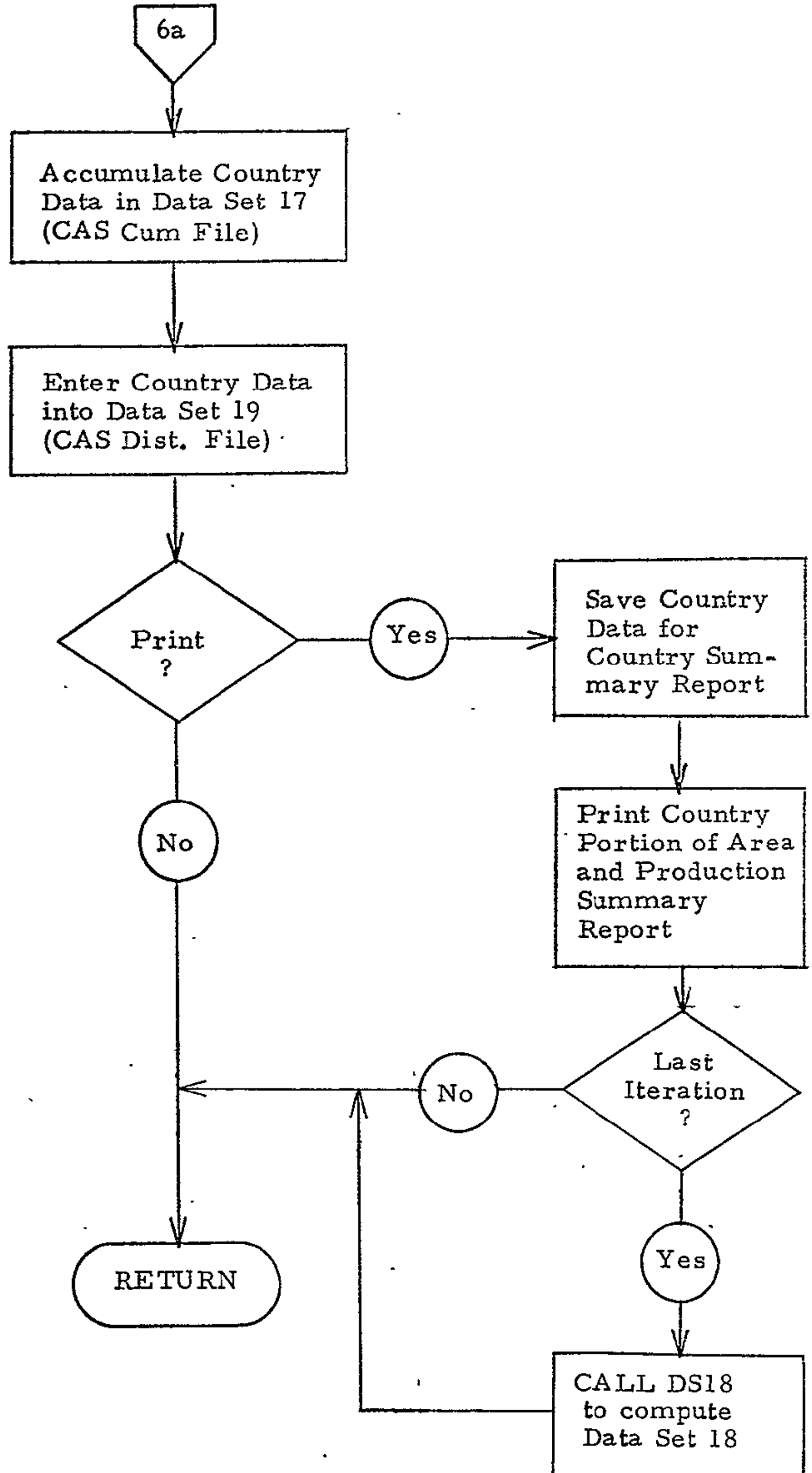
CAS3 Flow Diagram (Sheet 1 of 4)



CAS3 Flow Diagram (Sheet 2 of 4)



CAS3 Flow Diagram (Sheet 3 of 4)



CAS3 Flow Diagram (Sheet 4 of 4)

Subroutine GETYS

Purpose:

Subroutine GETYS reads strata yield data from the YES output file (YES) and obtains the proper value of estimated yield for the current bio-window or prediction date.

Input:

Quantity	Common Block	File	Source
PPFLG	CASFLG		CAS
IPD	CASFLG		CAS
PPDATE	CASFLG		CAS
NSTRAT	CASFLG		CASINL, DS456
NRYES	CASFLG		CASINL
NSTART	CNTRL		LEM
ENDFIL	CONST		Block Data
STRATA	DSET4	YESOUT	YES
EVYRS	DSET4	YESOUT	YES
ZONE	DSET7	YESOUT	YES
REGION	DSET8	YESOUT	YES
YS	DSET10	YESOUT	YES
ESTYS	DSET10	YESOUT	YES
YESOUT	FILES		Block Data
IREG2	SSHDTA	SUBHST	CASPP
IZONE2	SSHDTA	SUBHST	CASPP
ISTRA2	SSHDTA	SUBHST	CASPP
NT	STATS		LEM
YSTR	YESDTA	YESOUT	YES
IZPRDD	YESDTA	YESOUT	YES
YSCI	YESDTA	YESOUT	YES
YSYCI	YESDTA	YESOUT	YES
YCOUN	-	YESOUT	YES

Output:

Quantity	Common Block	Used By
NRYES	CASFLG	WRAPUP
STRATA	DSET4	CASPP, DS456
EVYRS	DSET4	DS456
ZONE	DSET7	CASPP, DS7
REGION	DSET8	CASPP
YS	DSET10	DS456
ESTYS	DSET10	DS456

Linkage:

CALL GETYS

Subroutines Used:

ERRMES

Local Variables:

I	-	Index in DO loop
II	-	I-6
YCOUN	-	Country ID read from YESOUT file

Processing:

1. Advance NRYES by 1 and read one data record from YESOUT file.

2. Check for end-of-data indicator (country ID = 4H ZZZZ). If end-of-data read, call ERRMES to write error message and abort run.

3. If first iteration of current run and if not first strata in country, compare region, zone and strata ID's from YESOUT and SUBHST files. If any pair does not agree, call ERRMES to write error message and abort run.

4. If PPFLG = 0 (bio-window), then find last non-zero yield date from the YESOUT file for the current strata. Save the corresponding value of estimated yield in ESTYS and save the corresponding value of yield variance in EVYRS. Then return.

5. If PPFLG $\neq 0$ (prediction date), then find the last non-zero yield date which is less than or equal to the given prediction date. Save the corresponding estimated yield in ESTYS and save the corresponding yield variance in EVYRS. If the given prediction date is less than all yield dates for this strata, then a flag is set so CAS will skip this strata.

6. Return.

Subroutine GROUP

Purpose:

Subroutine GROUP reads segment data from the CAMS output file (CAMSF), selects the estimated proportion wheat for the proper bio-window for each segment, and aggregates the segment data up to the substrata level.

Input:

Quantity	Common Block	File	Source
WPRIOR	CASCM		CASIN
PPFLG	CASFLG		CAS
IBW	CASFLG		CAS
PPDATE	CASFLG		CAS
NRCAMS	CASFLG		CASINL
M1K	DSET1		DS123
EPWK	DSET1		DS123
EPW2K	DSET1		DS123
SMPKPI	DSET1		DS123
SUMPK2	DSET1		DS123
SUMPK	DSET1		DS123
CAMSF	FILES		Block Data
IDSEGT	SEGDTA	CAMSF	CAMS
ISEG	SEGDTA	CAMSF	CAMS
TPWKI	SEGDTA	CAMSF	CAMS
ZACDAY	SEGDTA	CAMSF	CAMS
EPWKI	SEGDTA	CAMSF	CAMS
ERRPWI	SEGDTA	CAMSF	CAMS
IREG2	SSHDTA		CASPP
IZONE2	SSHDTA		CASPP
ISTRA2	SSHDTA		CASPP
ISUBS2	SSHDTA		CASPP
NSEG	SSHDTA		CASPP
HISTPW	SSHDTA		CASPP

Output:

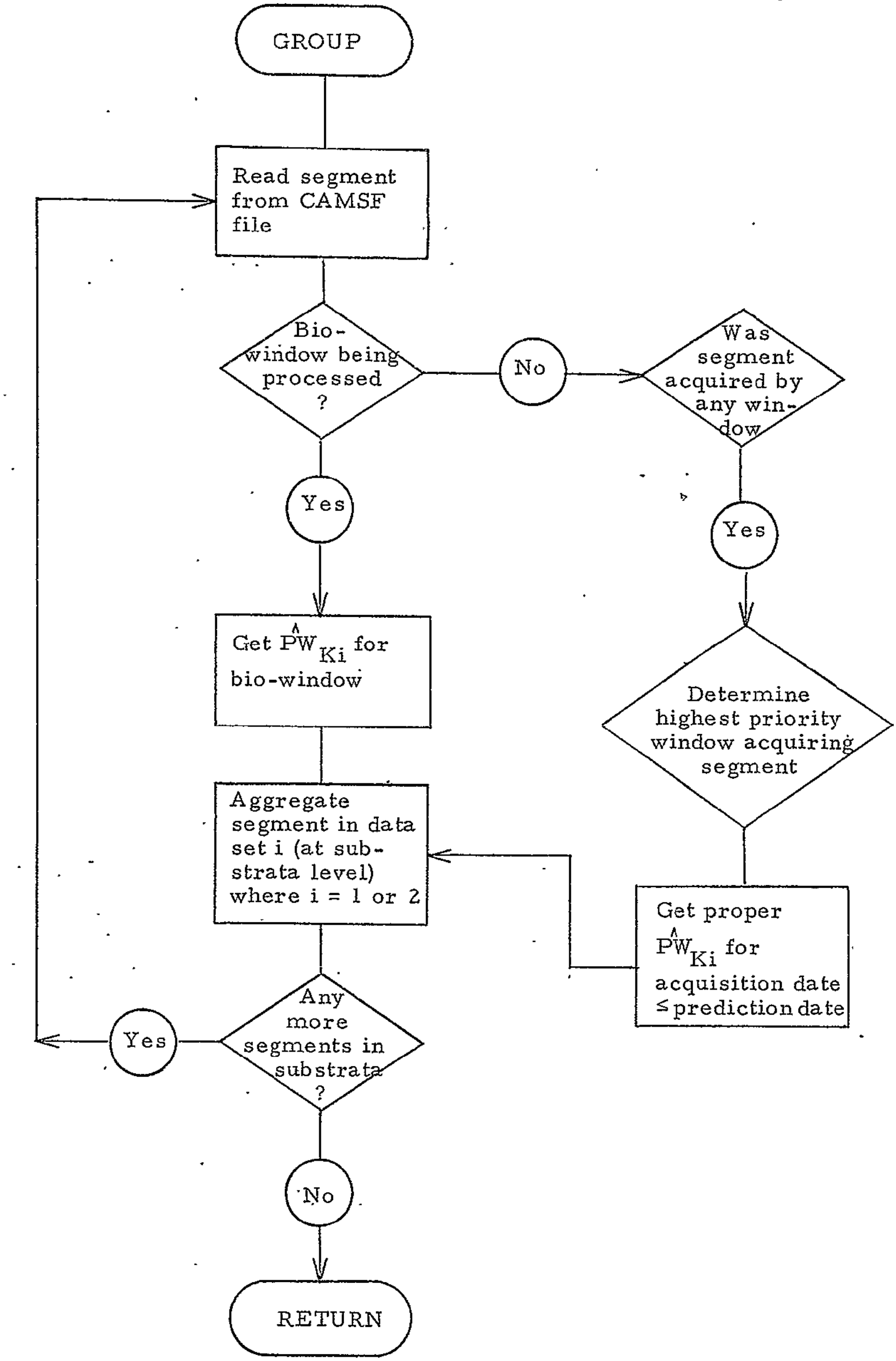
Quantity	Common Block	
M1K	DSET1	Values share same location; one or the other is computed.
M2K	DSET1	
EPWK	DSET1	
EPW2K	DSET1	
SMPKPI	DSET1	
SUMPK2	DSET1	
SUMPK	DSET1	

Linkage:

CALL GROUP

Local Variables:

I	-	Index in DO loop
N	-	Index in segment DO loop
ESTPWI	-	Specific value of estimated proportion wheat (fraction)


GROUP
Read segment from CAMSF file
Bio-window being processed ?
No
Was segment acquired by any window
Yes
Get $\hat{PW}_{Ki}$ for bio-window
Determine highest priority window acquiring segment
Aggregate segment in data set i (at sub-strata level) where i = 1 or 2
Get proper $\hat{PW}_{Ki}$ for acquisition date ≤ prediction date
Any more segments in substrata ?
Yes
No
RETURN

Subroutine SUMREP

Purpose:

Subroutine SUMREP prints the Country Summary Report, a two page printed report which specifies the mean values of the estimated wheat area, y eld, and production, coefficients of variance, and confidence levels at the country level for each bio-window and prediction date.

Input:

Quantity	Common Block	Source
OUTP	FILES	Block Data
CUNTRY	LEMCM	INPUT
MXLINE	PAGECM	Block Data
NT	STATS	LEM
AREACF	CASCM	CASIN
YCF	CASCM	CASIN
PRDCF	CASCM	CASIN
APRUTS	CASCM	CASIN
PPRUTS	CASCM	CASIN
YPRUTS	CASCM	CASIN
AUNITS	CASCM	CASIN
BWIND	CASCM	CASIN
IPRD	CASCM	CASIN
NPDATE	CASCM	CASIN
HWAC	DSET13	CAS2
TWAC	DSET13	CAS2
TPRODC	DSET13	CAS2
TYC	DSET13	CAS2
CSUMR	SUMDTA	CASOUT

Output:

1) Printed output

CUNTRY	CSUMR
NT	HWAC
APRUTS	TWAC
YPRUTS	TYC
PPRUTS	TPRODC
IPRD	

2) Output to common

Quantity	Common Block	Used By
NLINE	PAGECM	PAGER

Linkage:

CALL SUMREP

Subroutines Used:

EJECT
PAGER

Local Variables:

I - Index in DO loop, e.g., bio-window number
INDX = AUNITS + 1

Processing:

1. Call EJECT to eject page and print page headers.
2. Print labels for Country Summary Report.
3. Print country ID and Monte Carlo iteration number.
4. Print output labels (including units labels for first page only).
5. Print country summary data for each bio-window and each prediction point.

First Page

Bio-window or prediction	Est. WA	CV Anal. WA (Pct True)	CV Area Est. (Pct True)	CV Area Error (Pct True)
	Est. Yield	Std. Dev. Pct Error		
	Est. Prod.	CV Anal. Prod (Pct True)	CV Prod. Est. (Pct True)	CV Prod. Error (Pct True)

Second Page

Bio-window or prediction date	CL Area True/Error	CL Area Est/Est	CL Area True/Est	CL Area True/WC
	CL Prod. True/Error	CL Prod. Est/Est	CL Prod. True/Est	CL Area True/WC

6. At the bottom of the first page print

   Historical wheat area

   True wheat area

   True yield

   True production

Note: Steps 1-5 are performed for Page 1 and Page 2 of the Country Summary Report.

Subroutine TSUB

Purpose:

Subroutine TSUB computes the quantity T, which is the second term of the PPS area variance equation.

Input:

Quantity	Common Block	Source
XM2JS	DSET4	DS123
HWAS2	DSET4	DS123
HWA2K	CASCUM	DS123
WAKNEY	CASCUM	DS123
T	DSET4	DS456

Output:

Quantity	Common Block	Used By
T (updated value)	DSET4	DS456

Linkage:

CALL TSUB

Subroutines Used:

None.

Local Variables:

$$\text{CON} = \frac{M_{2j}}{\widetilde{WA}_{2S}}$$

$\text{M2J} = M_{2j}$ (integer)

PIK = Array of π_K

K = Substrata index

$$SUM2 = \sum_{\alpha=1}^{S2} \pi_\alpha^2$$

$$SUM3 = \sum_{\alpha=1}^{S2} \pi_\alpha^3$$

$$CON1 = \frac{M_{2j} - 1}{M_{2j}}$$

$$CON2 = \frac{M_{2j} - 1}{M_{2j}^2}$$

$$CON3 = 2\left(\frac{M_{2j} - 1}{M_{2j}^3}\right)$$

$$CON3S = \left(\frac{M_{2j} - 1}{M_{2j}^3}\right)\left(\sum_{\alpha=1}^{S2} \pi_\alpha^2\right)$$

$$CON4S = 3\left(\frac{M_{2j} - 1}{M_{2j}^4}\right)\left(\sum_{\alpha=1}^{S2} \pi_\alpha^2\right)$$

$$CON4S3 = 2\left(\frac{M_{2j} - 1}{M_{2j}^4}\right)\left(\sum_{\alpha=1}^{S2} \pi_\alpha^3\right)$$

$$CON5S2 = 3\left(\frac{M_{2j} - 1}{M_{2j}^5}\right)\left(\sum_{\alpha=1}^{S2} \pi_\alpha^2\right)^2$$

$$M2JM1 = M_{2j} - 1$$

$$WAKPIK = \frac{WA'_K}{\pi_K}$$

$$PIK2 = \pi_K^2$$

$$PIK3 = \pi_K^3$$

KP1 = K + 1

PIKPKP = $\pi_K \; \pi_{K'}$

TERM2 = $\pi_K^{\;2} \; \pi_{K'} + \pi_K \; \pi_{K'}^{\;2}$

PIKPP = $\pi_{K''}$

Processing:

1. Compute

$$\text{PIK(K)} = \pi_K = \text{XM2JS}\left[\frac{\text{HWA2K(K)}}{\text{HWAS2}}\right] = M_{2j}\left[\frac{\widetilde{WA}_{2K}}{\widetilde{WA}_{2S}}\right]$$

for each group II substrata in the stratum.

2. Compute SUM2 $= \sum_{\alpha=1}^{S2} \pi_\alpha^{\;2}$ and SUM3 $= \sum_{\alpha=1}^{S2} \pi_\alpha^{\;3}$

3. Compute coefficients CON1, CON2, CON3, CON3S, CON4S, CON4S3, CON5S2

4. Compute for each pair of distinct group II substrata

   K, KP = K' $\neq$ K

   the following

   a) PIKPKP = PIK(K) * PIK(KP)
      $= \pi_K * \pi_{K'}$

   b) TERM2 = $\pi_K^{\;2} \; \pi_{K'} + \pi_K \; \pi_{K'}^{\;2}$

   c) PIKPP = $\pi_{K''}$

      = CON1 * PIKPKP + CON2 * TERM2

      - CON3S * PIKPKP

      + CON3 * [PIK(K)^3 PIK(KP) + PIK(K) * PIK(KP)^3
      + PIK(K)^2 * PIK(KP)^2]

      - CON4S * TERM2 + CON5S2 * PIKPKP
      - CON4S3 * PIKPKP

5. Compute

$$T = \sum_{K=1}^{S2-1} \sum_{K'=K+1}^{S2} (\pi_K \, \pi_{K'} - \pi_{K''}) \left(\frac{WA'_K}{\pi_K} - \frac{WA'_{K'}}{\pi_{K'}} \right)$$

where

WA'_K and $WA'_{K'}$ are the non-epoch wheat areas for substrata K and K'.

SUBROUTINE CLASSN

Purpose:

This routine controls the computation of the class number for each substrata in SUBHST file as defined by STARTR-ENDZ user inputs. The SUBHST file is read a zone at a time for the first prediction point and a new temporary file ISUBH2 is generated containing all necessary SUBHST data. For subsequent prediction points the ISUBH2 file is read instead of SUBHST. For all prediction points a strata table is formed and then this routine controls the computation of the class number. It then updates the ISUBH2 file with the class numbers for the appropriate prediction point.

Input:

CASFLG COMMON:
  H, NRSSH, LDS1, IPP, NCAMSK
SSHDTA COMMON:
  All data except CLASS
CONST COMMON:
  ENDFIL
LEMCM COMMON:
  ENDR, ENDZ
IXSUBH COMMON:
  L1XSSH, IXSUBH
FILES COMMON:
  LSUBH, ISUBH2, CAMSF, OUTP
DSET1 COMMON:
  M1K ≡ M2K
ARGLST COMMON:
  NFATAL
CLSTAB COMMON:
  IXPT, IBPT, IEPT

Output:

CLSTAB COMMON: ISUB1, NACQ

ISTRAT, ISBSTR, NSCNT, IGROUP, IDAT1, IDAT2

SSHDTA COMMON:

CLASS, MXK, VMULTK

CASFLG COMMON:

NRCAMS

Linkage:

CALL CLASSN

Subroutines Used:

CALL GROUP

CALL SEGTAB

CALL DETCLS

CALL ASSCLS (IOPT)

IOPT = 1 - Means all substrata in a zone are class 0

IOPT = 2 - Means all substrata in a zone are class 1

IOPT = 0 - Means that class numbers are to be assigned via computation

Subroutine RANACF is used to read/write file ISUBH2 as follows:

CALL RANACF (ISUBH2, 0, 0, 0, IXSUBH, LIXSSH, 0)
- Open file

CALL RANACF (ISUBH2, ISUB, SSHDTA, LSUBH2, IXSUBH, LIXSSH, n)
n = 1 - Read file
n = 2 - Write file

Local Variable Description:

ISUB - Count on number of records written/read on ISUBH2 file

IFIRST - First time flag, = 0 - not first time
$\neq$ 0 - first time

MAXSCT - Maximum substrata that can be handled in a zone, $\equiv$ 300

IBUF(39) - Array for temporary location of SUBHST or ISUBH2 when in read mode

NORD - Don't read flag = 0 - read; $\neq$ 0 - don't read

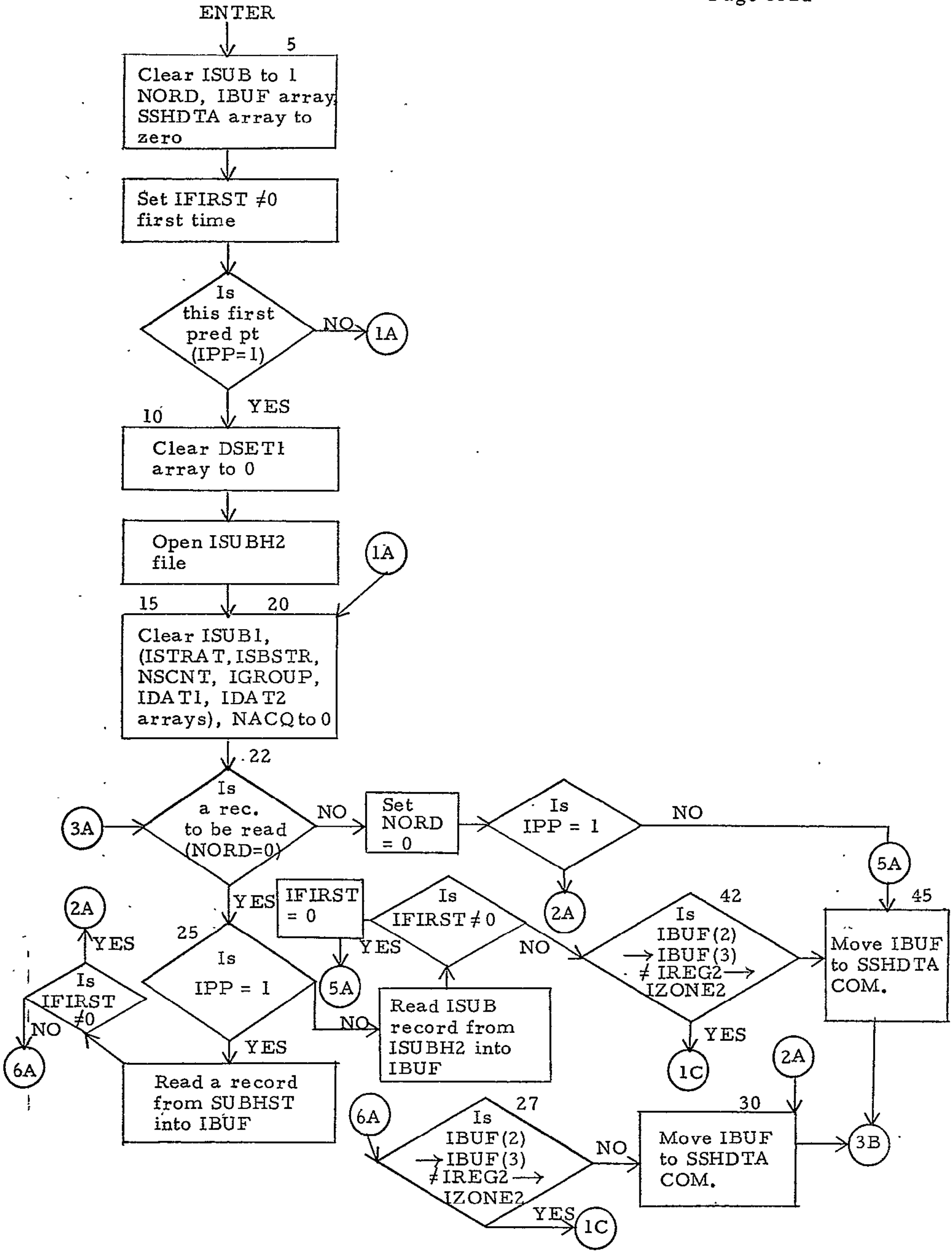
Processing:

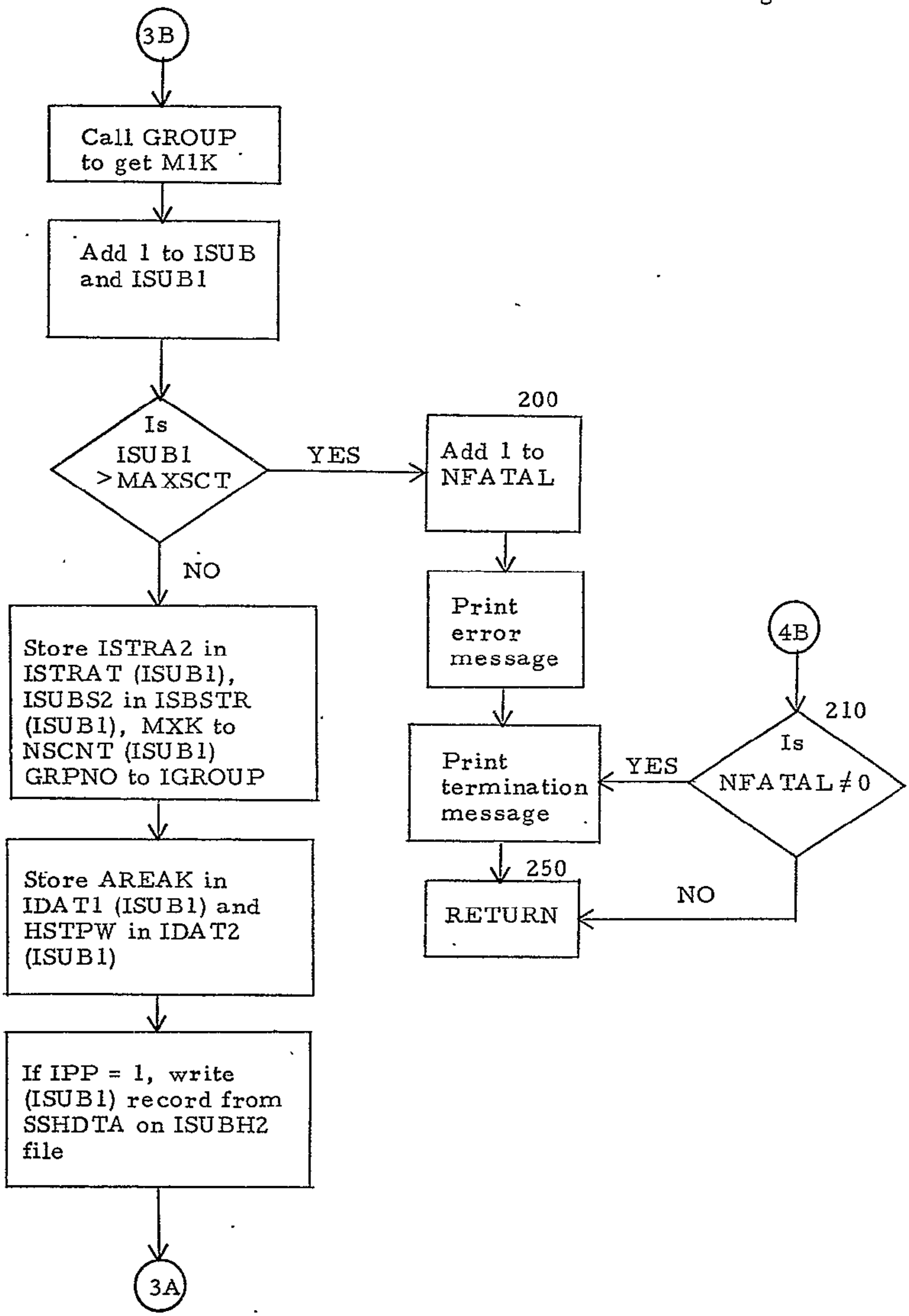
See flowchart for details.

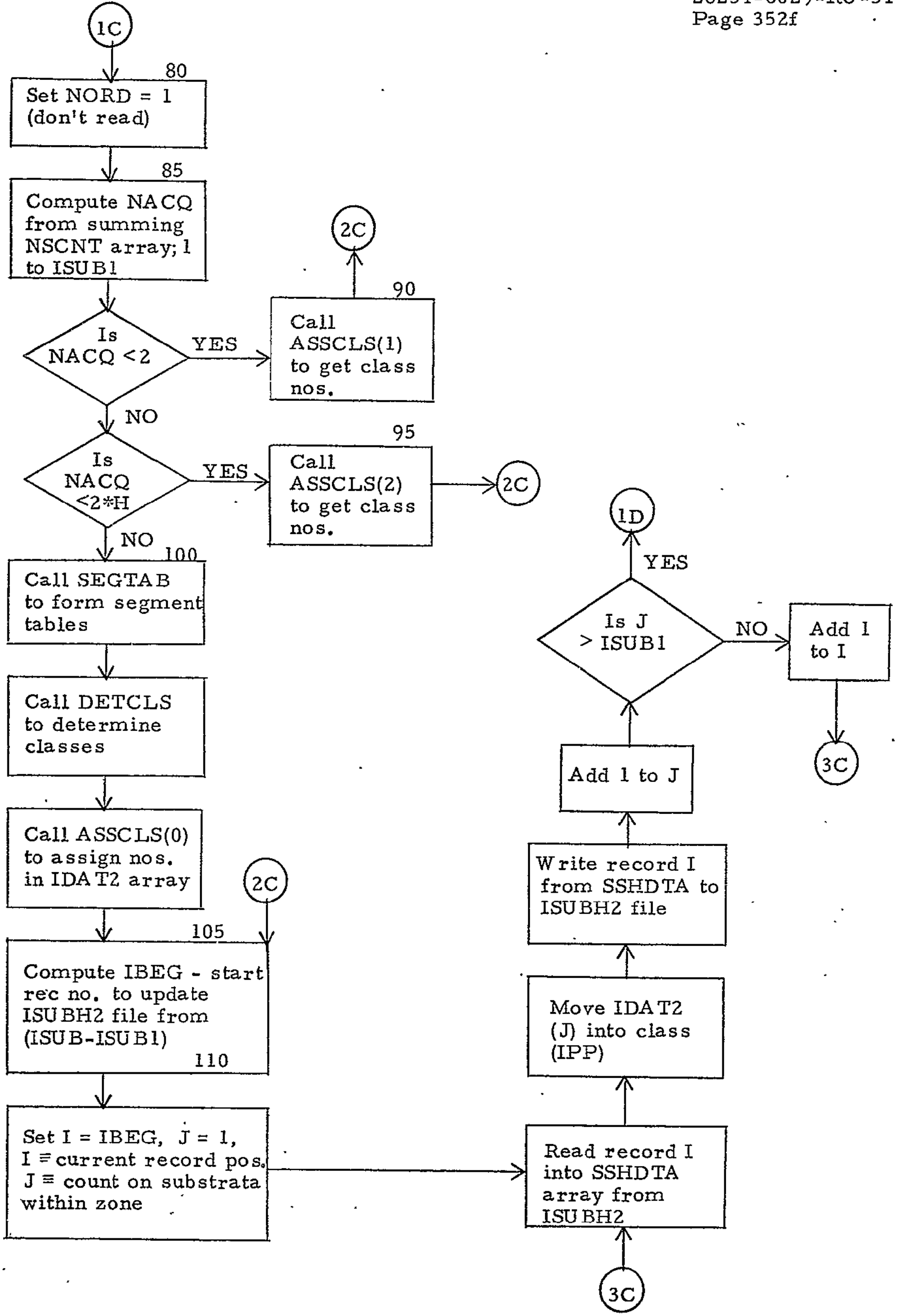
1. If NACQ is < 2, then calls to SEGTAB and DETCLS are not made and call ASSCLS(0) is made. Means not enough acquired segments.

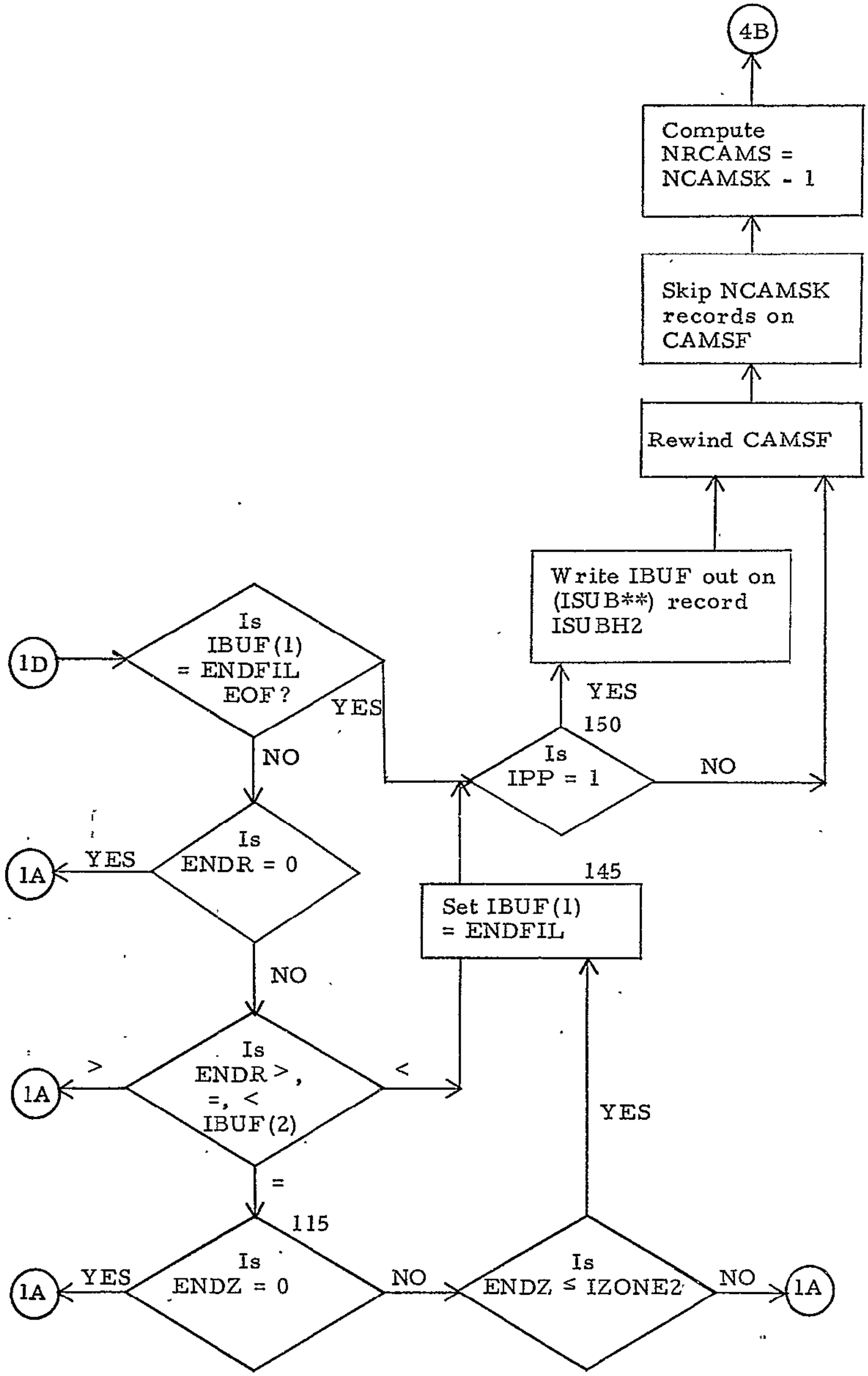
2. If NACQ is < 2. *H, then same as above with a call ASSCLS(1). Means only one class = 1 can be assigned. Not enough X's to split.

3. The CAMS file must be repositioned back to where it was upon entry to this routine. Subroutine GROUP controls the reading of CAMS.


ENTER
5
Clear ISUB to 1 NORD, IBUF array, SSHDTA array to zero
Set IFIRST ≠0 first time
Is this first pred pt (IPP=1)
NO
1A
YES
10
Clear DSET1 array to 0
Open ISUBH2 file
1A
15
20
Clear ISUB1, (ISTRAT, ISBSTR, NSCNT, IGROUP, IDAT1, IDAT2 arrays), NACQ to 0
22
3A
Is a rec. to be read (NORD=0)
NO
Set NORD = 0
Is IPP = 1
NO
5A
2A
YES
IFIRST = 0
Is IFIRST ≠ 0
YES
NO
5A
42
Is IBUF(2) → IBUF(3) ≠ IREG2 → IZONE2
45
Move IBUF to SSHDTA COM.
2A
YES
25
Is IPP = 1
NO
Read ISUB record from ISUBH2 into IBUF
YES
1C
Is IFIRST ≠0
NO
6A
Read a record from SUBHST into IBUF
2A
6A
27
Is IBUF(2) → IBUF(3) ≠ IREG2 → IZONE2
NO
30
Move IBUF to SSHDTA COM.
3B
YES
1C


3B
Call GROUP to get M1K
Add 1 to ISUB and ISUB1
Is ISUB1 >MAXSCT
YES
NO
200
Add 1 to NFATAL
Print error message
Store ISTRA2 in ISTRAT (ISUB1), ISUBS2 in ISBSTR (ISUB1), MXK to NSCNT (ISUB1) GRPNO to IGROUP
4B
210
Is NFATAL ≠ 0
YES
Print termination message
250
RETURN
NO
Store AREAK in IDAT1 (ISUB1) and HSTPW in IDAT2 (ISUB1)
If IPP = 1, write (ISUB1) record from SSHDTA on ISUBH2 file
3A


1C
80
Set NORD = 1 (don't read)
85
Compute NACQ from summing NSCNT array; 1 to ISUB1
Is NACQ < 2
YES
90
Call ASSCLS(1) to get class nos.
2C
NO
Is NACQ <2*H
YES
95
Call ASSCLS(2) to get class nos.
2C
NO
100
Call SEGTAB to form segment tables
Call DETCLS to determine classes
Call ASSCLS(0) to assign nos. in IDAT2 array
2C
105
Compute IBEG - start rec no. to update ISUBH2 file from (ISUB-ISUB1)
110
Set I = IBEG, J = 1,
I ≡ current record pos.
J ≡ count on substrata within zone
Read record I into SSHDTA array from ISUBH2
3C
Move IDAT2 (J) into class (IPP)
Write record I from SSHDTA to ISUBH2 file
Add 1 to J
Is J > ISUB1
YES
1D
NO
Add 1 to I
3C


4B
Compute NRCAMS = NCAMSK - 1
Skip NCAMSK records on CAMSF
Rewind CAMSF
Write IBUF out on (ISUB**) record ISUBH2
YES
150
Is IPP = 1
NO
1D
Is IBUF(1) = ENDFIL EOF?
YES
NO
1A
YES
Is ENDR = 0
NO
145
Set IBUF(1) = ENDFIL
1A
>
Is ENDR >, =, < IBUF(2)
<
=
YES
115
1A
YES
Is ENDZ = 0
NO
Is ENDZ ≤ IZONE2
NO
1A

SUBROUTINE SEGTAB

Purpose:

Given a set of data for one zone in a strata table set, this routine computes a normalized array of standard deviations, X_i, sorts them in ascending order, computes the gaps between the sorted X's and ranks them. This data is output in the segment tables.

Input:

CLSTAB COMMON:

NSCNT, IDAT1, IDAT2, ISUB1

Output:

CLSTAB COMMON:

IDAT1, XORD, IXPT, IRANK

Linkage:

CALL SEGTAB

Subroutines Used:

CALL SORTAG (IRANK, 1, IPT, IXPT)

Local Variable Description:

IPT - Number of items in segment arrays = sum of all NSCNT's in zone

GAP(300) - Table of gap values between sorted X_i for each substrata in a zone ≡ GAP

SUM - $\sum_{I=1}^{IPT} T(I)$

ICON - 10^{20}

XMIN - Current minimum value in gap array

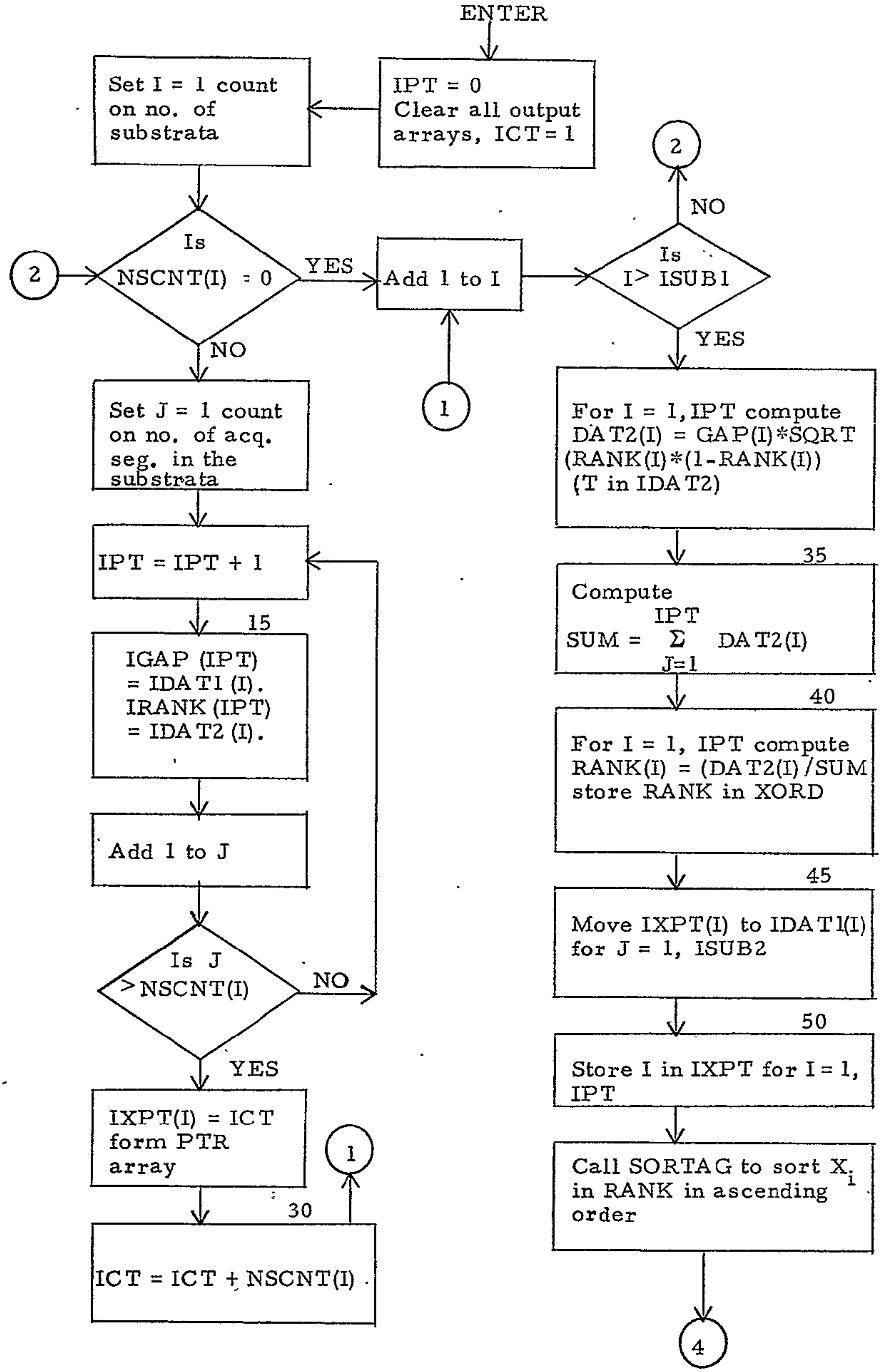
XMINS - Saved minimum value from gap array

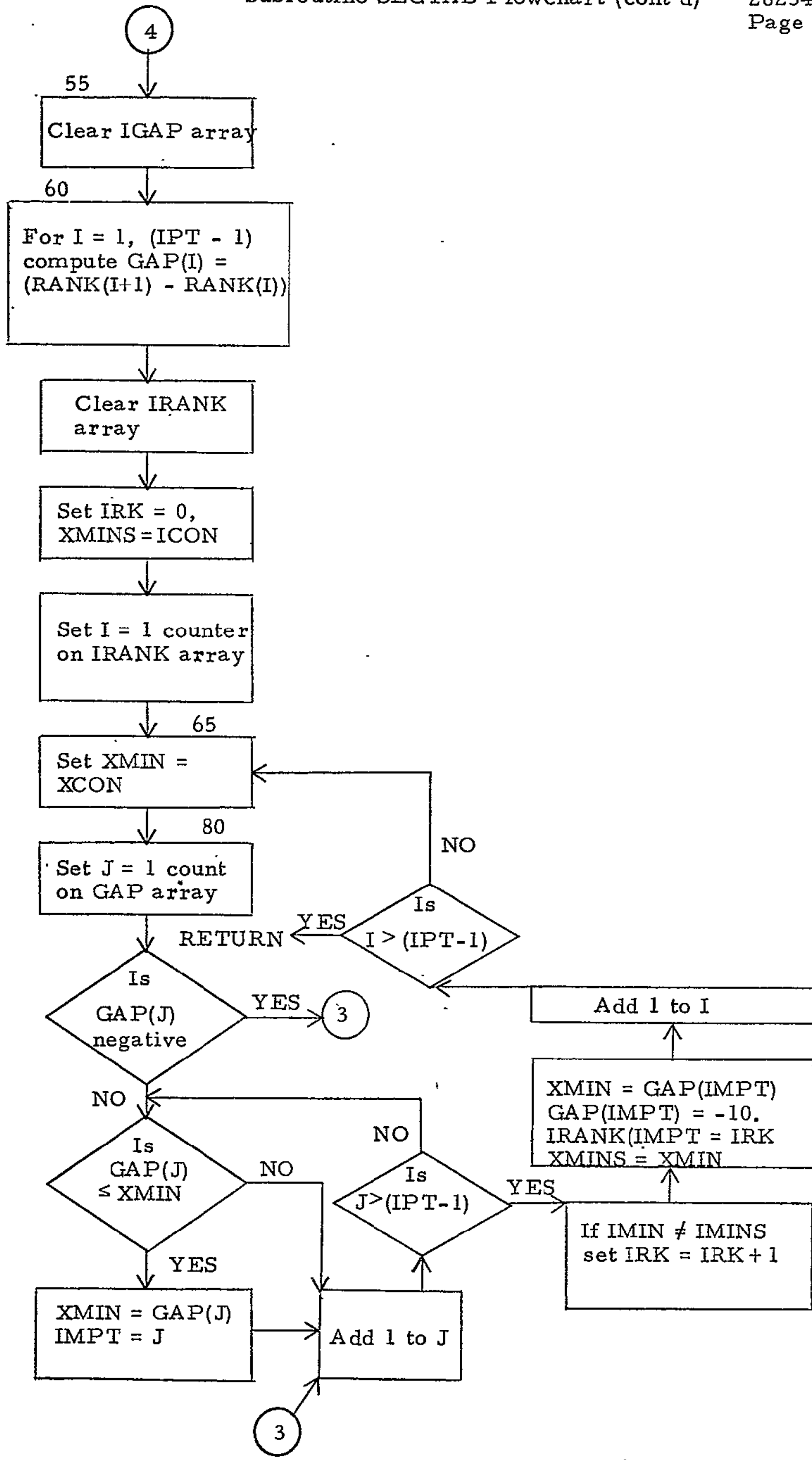
IRK - Current rank value (1-N)

IMPT - Subscript in GAP of current smallest value

Processing:

See flowchart.


ENTER
IPT = 0 Clear all output arrays, ICT = 1
Set I = 1 count on no. of substrata
Is NSCNT(I) = 0
YES
NO
Add 1 to I
Is I> ISUB1
NO
YES
Set J = 1 count on no. of acq. seg. in the substrata
IPT = IPT + 1
15
IGAP (IPT) = IDAT1 (I). IRANK (IPT) = IDAT2 (I).
Add 1 to J
Is J >NSCNT(I)
NO
YES
IXPT(I) = ICT form PTR array
30
ICT = ICT + NSCNT(I)
For I = 1, IPT compute DAT2(I) = GAP(I)*SQRT (RANK(I)*(1-RANK(I)) (T in IDAT2)
35
Compute SUM = Σ DAT2(I) IPT J=1
40
For I = 1, IPT compute RANK(I) = (DAT2(I)/SUM store RANK in XORD
45
Move IXPT(I) to IDAT1(I) for J = 1, ISUB2
50
Store I in IXPT for I = 1, IPT
Call SORTAG to sort X i in RANK in ascending order
1
2
4


4
55
Clear IGAP array
60
For I = 1, (IPT - 1) compute GAP(I) = (RANK(I+1) - RANK(I))
Clear IRANK array
Set IRK = 0, XMINS = ICON
Set I = 1 counter on IRANK array
65
Set XMIN = XCON
80
Set J = 1 count on GAP array
Is GAP(J) negative
YES
3
NO
Is GAP(J) ≤ XMIN
NO
YES
XMIN = GAP(J) IMPT = J
Add 1 to J
3
Is J>(IPT-1)
NO
YES
If IMIN ≠ IMINS set IRK = IRK+1
XMIN = GAP(IMPT) GAP(IMPT) = -10. IRANK(IMPT = IRK XMINS = XMIN
Add 1 to I
Is I > (IPT-1)
YES
RETURN
NO

SUBROUTINE DETCLS

Purpose:

This routine determines how the array XORD is to be broken down into classes. It produces the count of classes and the beginning and ending subscript in XORD for each class.

Input:

CLSTAB COMMON:

XORD, IXPT, MAXCLS, IRANK, NACQ, IXPT

CASFLG COMMON:

H ≡ IH

Output:

CLSTAB COMMON:

IBPT, IEPT and ICLNT

Linkage:

CALL DETCLS

Subroutines Used:

CALL SORTAG (IDT, 1, ICLNT, IDUM)

Local Variable Description:

CC - Constant ≡ .25 minimum allowed gap within a class of X's

K - Counter on number of rank entries

I - Counter within rank table

ID(10) - Table of class breakpoints in XORD

IDUM(10) - Pointer into ID

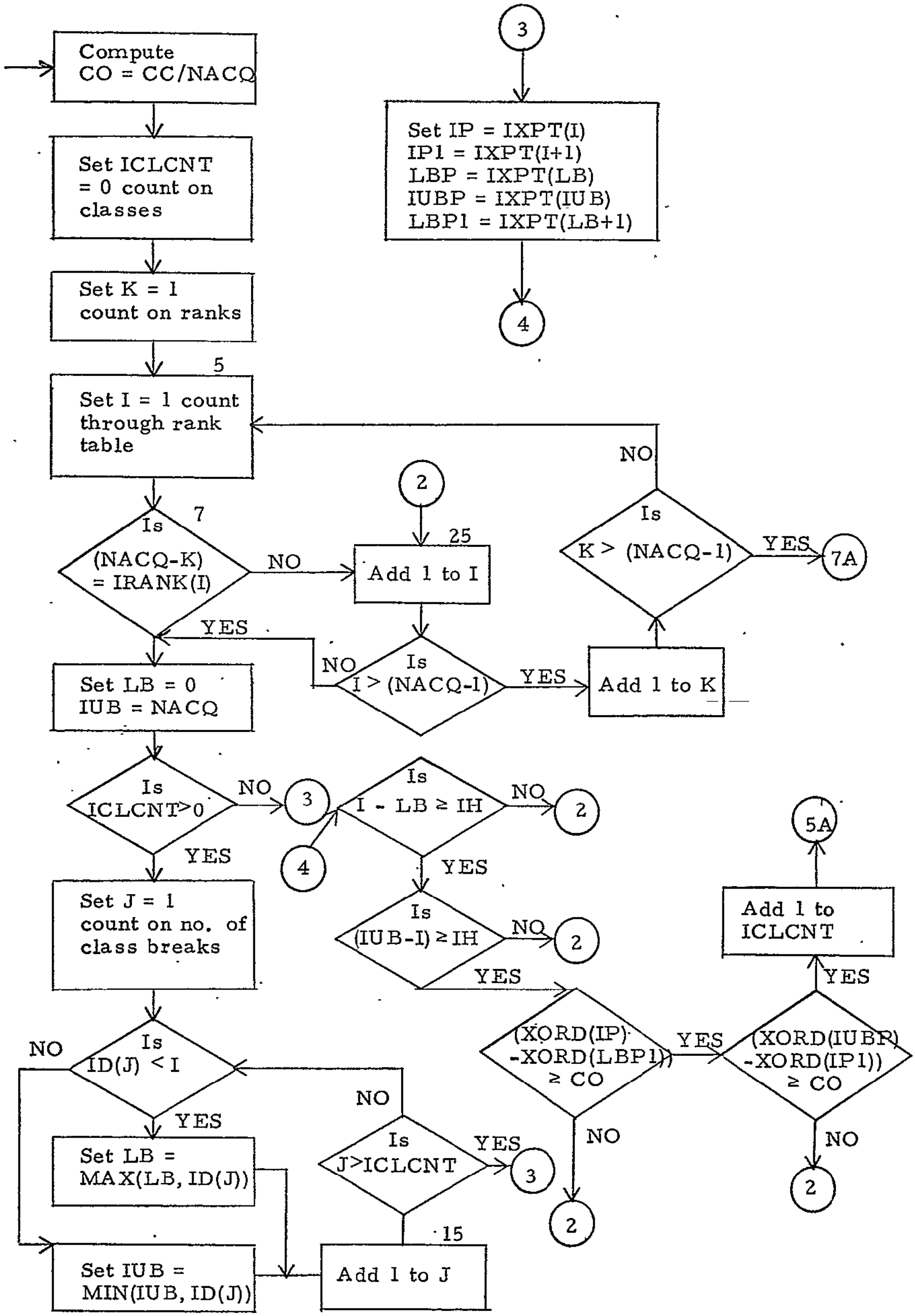
LB - Low boundary to search for class break

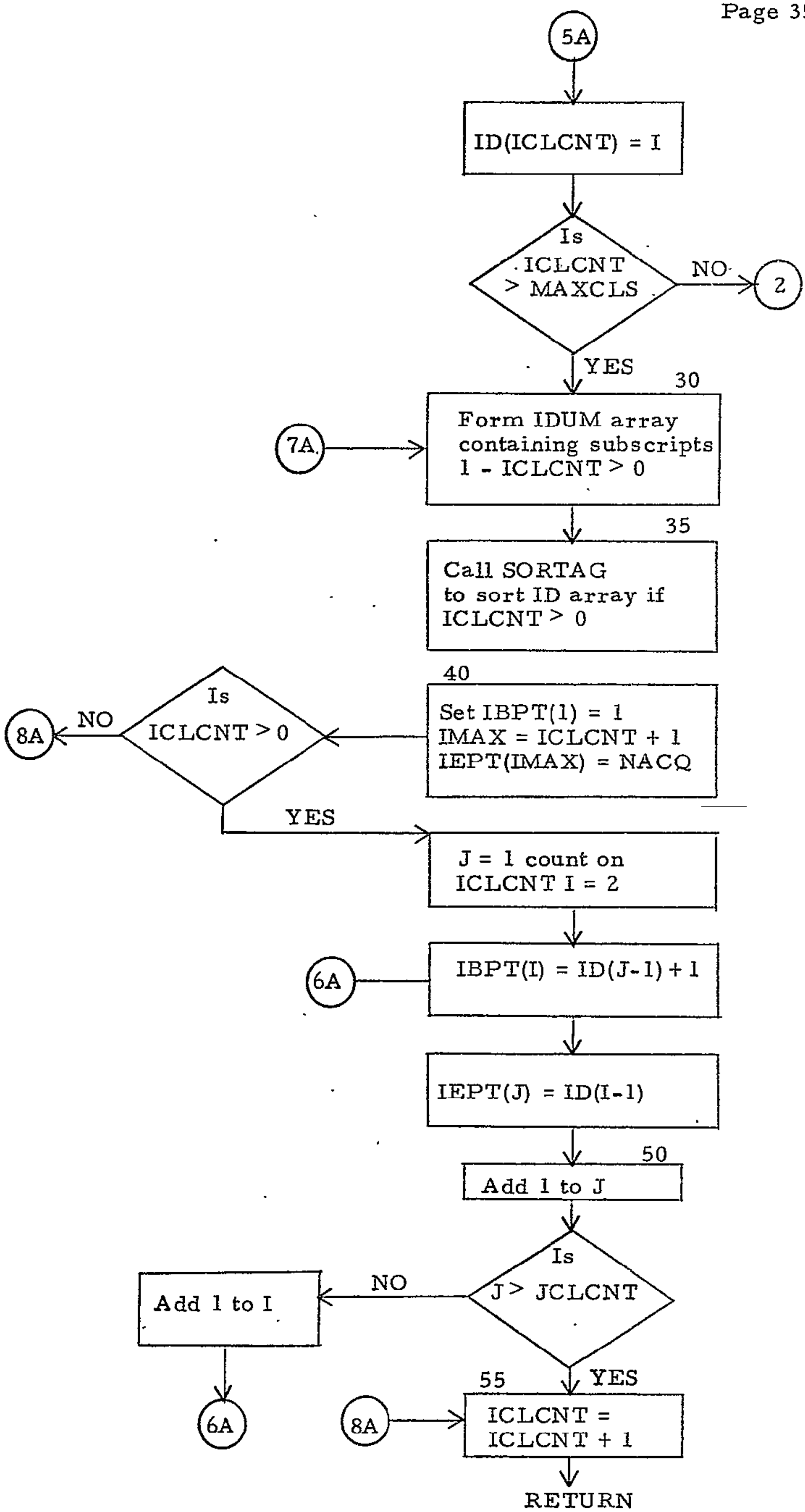
IUB - Upper boundary to search for class break

J - Count on class breaks

Processing:

See flowchart.


Compute
CO = CC/NACQ
Set ICLCNT
= 0 count on
classes
Set K = 1
count on ranks
5
Set I = 1 count
through rank
table
7
Is
(NACQ-K)
= IRANK(I)
NO
25
Add 1 to I
YES
NO
Is
I > (NACQ-1)
YES
Add 1 to K
Is
K > (NACQ-1)
YES
7A
NO
3
Set IP = IXPT(I)
IP1 = IXPT(I+1)
LBP = IXPT(LB)
IUBP = IXPT(IUB)
LBP1 = IXPT(LB+1)
4
Set LB = 0
IUB = NACQ
Is
ICLCNT>0
NO
3
YES
Is
I - LB ≥ IH
NO
2
4
YES
Is
(IUB-I) ≥ IH
NO
2
YES
(XORD(IP)
-XORD(LBP1))
≥ CO
YES
(XORD(IUBP)
-XORD(IP1))
≥ CO
YES
Add 1 to
ICLCNT
5A
NO
2
NO
2
Set J = 1
count on no. of
class breaks
Is
ID(J) < I
NO
YES
Set LB =
MAX(LB, ID(J))
Set IUB =
MIN(IUB, ID(J))
NO
Is
J>ICLCNT
YES
3
15
Add 1 to J


5A
ID(ICLCNT) = I
Is ICLCNT > MAXCLS
NO
2
YES
30
7A
Form IDUM array containing subscripts 1 - ICLCNT > 0
35
Call SORTAG to sort ID array if ICLCNT > 0
40
Set IBPT(1) = 1
IMAX = ICLCNT + 1
IEPT(IMAX) = NACQ
Is ICLCNT > 0
NO
8A
YES
J = 1 count on ICLCNT I = 2
6A
IBPT(I) = ID(J-1) + 1
IEPT(J) = ID(I-1)
50
Add 1 to J
Is J > JCLCNT
NO
Add 1 to I
6A
YES
55
8A
ICLCNT = ICLCNT + 1
RETURN

SUBROUTINE ASSCLS

Purpose:

To assign class numbers to all substrata defined by ISTRAT and ISUBST in the strata tables.

Input:

CLSTAB COMMON:

ISTRAT, ISUBST, NSCNT, IGROUP, IDAT1, IXPT, IBPT, IEPT, ICLCNT, NACQ, ISUB1

Output:

CLSTAB COMMON:

IDAT2

Linkage:

CALL ASSCLS (IOPT)

IOPT is input as follows:

IOPT = 0 - Compute class numbers using the algorithm

IOPT = 1 - Set all class numbers to 0

IOPT = 2 - Set all class numbers to 1

Subroutines Used:

None.

Local Variable Description:

ISTART, IEND - Group of substrata with = strata ID

ISTRSV - Saved strata to see if new strata occurs

ICT(10) - Count of substrata for each class

IFLAG - = 0 - No substrata within strata has segm.
$\neq$ 0 - At least 1

IFLAG1 - = 0 - not done, $\neq$ 0 - last entry in IGROUP passed

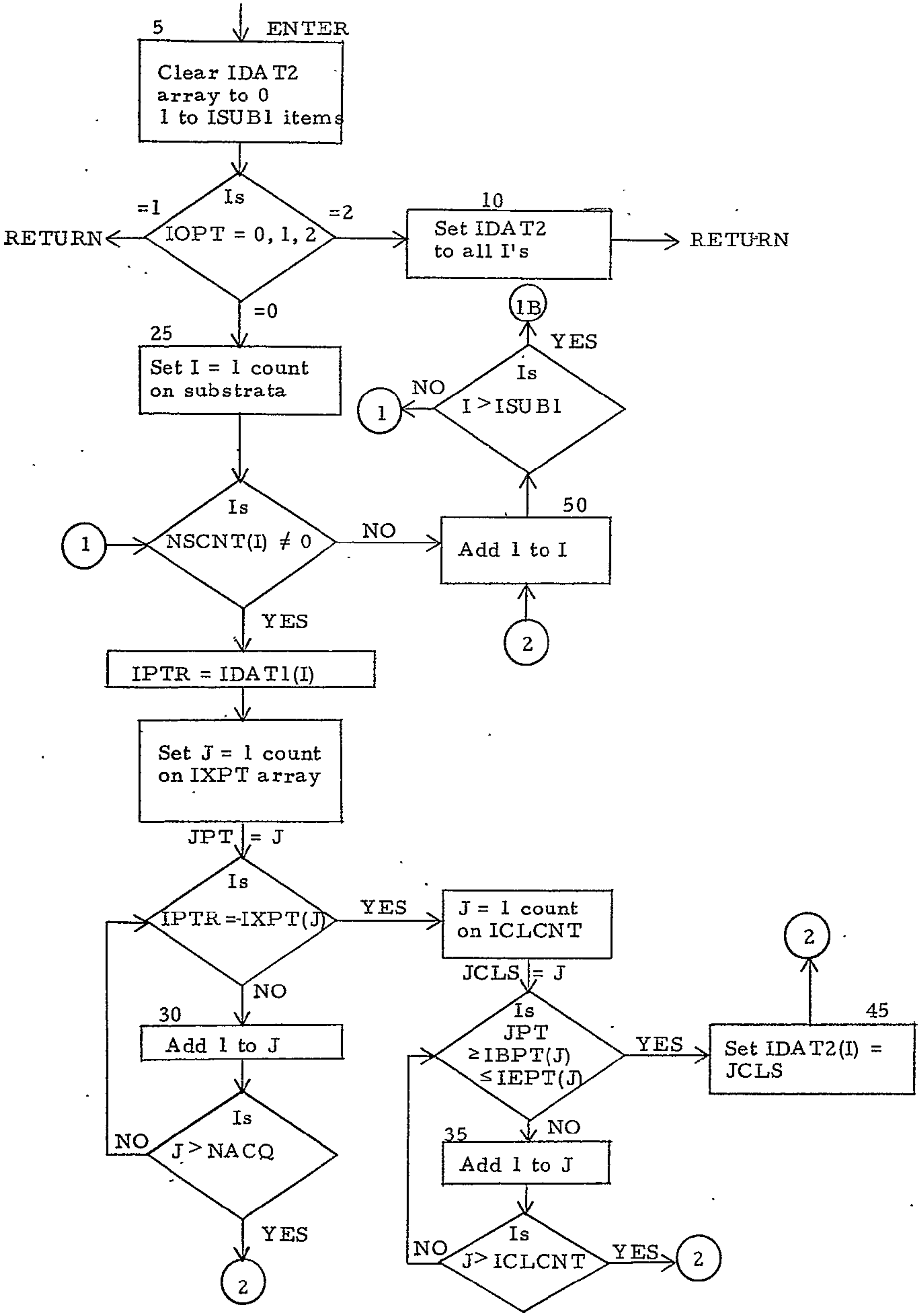
IJCL - Class number with most substrata

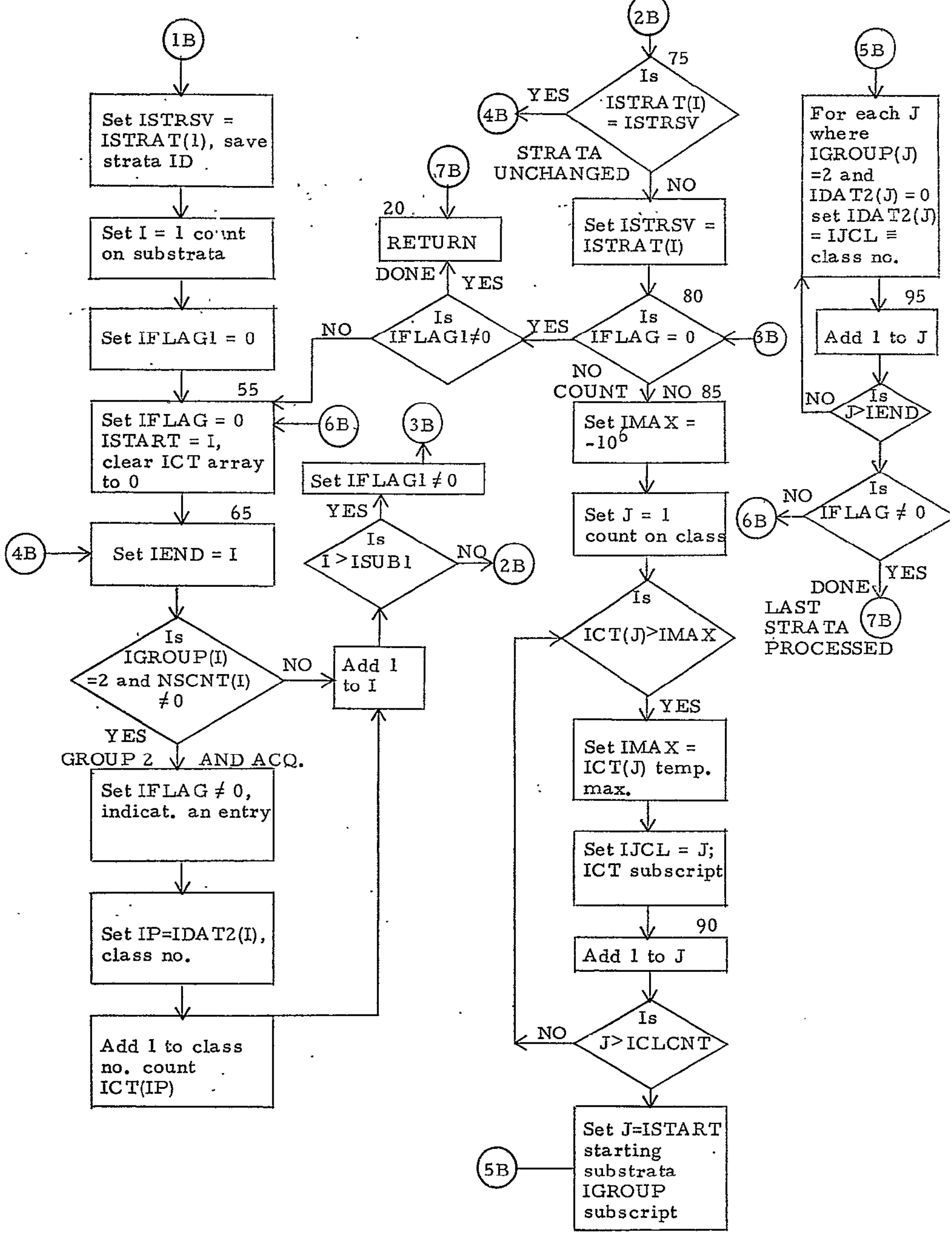
IMAX - Largest substrata count

Processing:

See flowchart.

Subroutine ASSCLS Flowchart




1B
Set ISTRSV = ISTRAT(1), save strata ID
Set I = 1 count on substrata
Set IFLAG1 = 0
55
Set IFLAG = 0 ISTART = I, clear ICT array to 0
6B
65
4B
Set IEND = I
Is IGROUP(I) =2 and NSCNT(I) ≠ 0
NO
YES
GROUP 2 AND ACQ.
Set IFLAG ≠ 0, indicat. an entry
Set IP=IDAT2(I), class no.
Add 1 to class no. count ICT(IP)
Add 1 to I
Is I > ISUB1
NO
2B
YES
Set IFLAG1 ≠ 0
3B
7B
20
RETURN
DONE
YES
Is IFLAG1≠0
NO
YES
2B
75
Is ISTRAT(I) = ISTRSV
YES
4B
STRATA UNCHANGED
NO
Set ISTRSV = ISTRAT(I)
80
Is IFLAG = 0
3B
NO COUNT
NO 85
Set IMAX = -10^6
Set J = 1 count on class
Is ICT(J)>IMAX
YES
Set IMAX = ICT(J) temp. max.
Set IJCL = J; ICT subscript
90
Add 1 to J
Is J>ICLCNT
NO
Set J=ISTART starting substrata IGROUP subscript
5B
5B
For each J where IGROUP(J) =2 and IDAT2(J) = 0 set IDAT2(J) = IJCL ≡ class no.
95
Add 1 to J
Is J>IEND
NO
Is IFLAG ≠ 0
NO
6B
YES
DONE
LAST STRATA PROCESSED
7B

YES SUBROUTINE DESCRIPTIONS

SUBROUTINE YES

Purpose:

Given the true yield for each strata, and also an error estimation of the bias and standard deviation of error, YES generates an estimated yield at from 1 to 6 estimation points in a simulation season, for the strata. An option allows the estimated yields to be the same as the true yields, bypassing the error simulation. A printed report is also optional.

Input:

1. YES is passed these quantities by LEM through COMMON (see LEM COMMON block descriptions for complete description of format):

   a. COMMON /LEMCM/

TITLE	Used by PAGER and EJECT
ICASE	Case no. of output file
STARTR ENDR STARTZ ENDZ	Start and ending regions and zones to process
IYES	=0, 1 do error simulation for estimation yields =3 bypass error simulation

   b. COMMON /CNTRL/

PRINTF	=0 no printed report =1 print report
SEED	SEED(5) is used in generating a random number for error simulation

   c. COMMON /FILES/

YESOUT LYESO YESERR LYESER	I/O device no's and no. of words in one record for YES input and output files
OUTP	Used in writing printed report

   d. COMMON /STATS/

ITER	Iteration no. for report

2. YES has one input file, YESERR, generated by the SEE program. Each record is read into a COMMON block. See Section 2.4 of the Users Manual for a description of the input file, and Programmers Manual for a detailed description of the COMMON block.

   a. COMMON /YESIN/

COUN	Country
IREG	Region
IZONE	Zone
ISTRAT	Strata
YTRUE	True yield
IZULU(6), BIAS(6), SD(6)	The Zulu date, bias, and standard deviation of error for each of up to six prediction points

Output:

1. YES passes these quantities back to the calling program LEM through COMMON (see COMMON block descriptions, for complete description of format):

   a. COMMON /CNTRL/

SEED	SEED(5) contains the seed for the random no. after the last use of it

   b. COMMON /STATS/

NREC	NREC(5) contains the no. of records processed from the YES input file (excluding header, records skipped, trailer)
NYESR	Contains the no. of records written onto the YES output file (excluding header and trailer)

2. YES produces one output file, YESOUT, for use by the program CAMS. Each record is written from the COMMON block. See Section 2.4 of the Users Manual for a description of the output file, and Programmers Manual for a detailed description of the COMMON block.

a. COMMON /YESOUT/

CID	Country
IREGID	Region
IZONID	Zone
ISTRID	Strata
YSTR	True yield
IZPRDD(6)	Zulu prediction date
YSCI(6)	Estimated yield
VSYCI(6)	Standard deviation of yield error

3. YES produces a printed report, on option. See YES Problem Description, Figure 2.

Linkage:

CALL YES — YES is called by the LEM program.

Subroutines Used:

CALL BETAD (SEED(5), 0, 0, RN, 1, IER) to get a random number RN from a normal distribution

CALL PAGER (NO) to print line on report

CALL EJECT (NO) to start new page on report

CALL FZULU (IZULU(J), IOUT) to convert Zulu date

CALL ERRMES (3HYES, 3HYES, 1, 1) to process error message

Local Variable Description:

IFILL	0 fill for header and trailer records
YNAME(2)	Output file name 3HYES
RN	Random number
IER	Error flag from BETAD (always 0)
ITEMP	No. of 0 fill words to put in record
IEND	Flag for end zone =0 haven't reached end zone yet =1 found end zone
INEW	Count of no. of strata per report page = 3 max.
ER	Used to compute % error; =0 unless true yield = 0, then = 10^{-6}

Processing:

See flow diagram for a flowchart of YES. The two equations used are:

a. YSCI = YSTR + BIAS(J) + RN * SD(J)

where		
YSCI	=	yield estimate for J^{th} prediction point
BIAS(J)	=	bias error for J^{th} prediction point for strata
YSTR	=	true yield of strata
RN	=	random number from a normal distribution
SD(J)	=	standard deviation of error at J^{th} point
J	=	integer, range 1-6

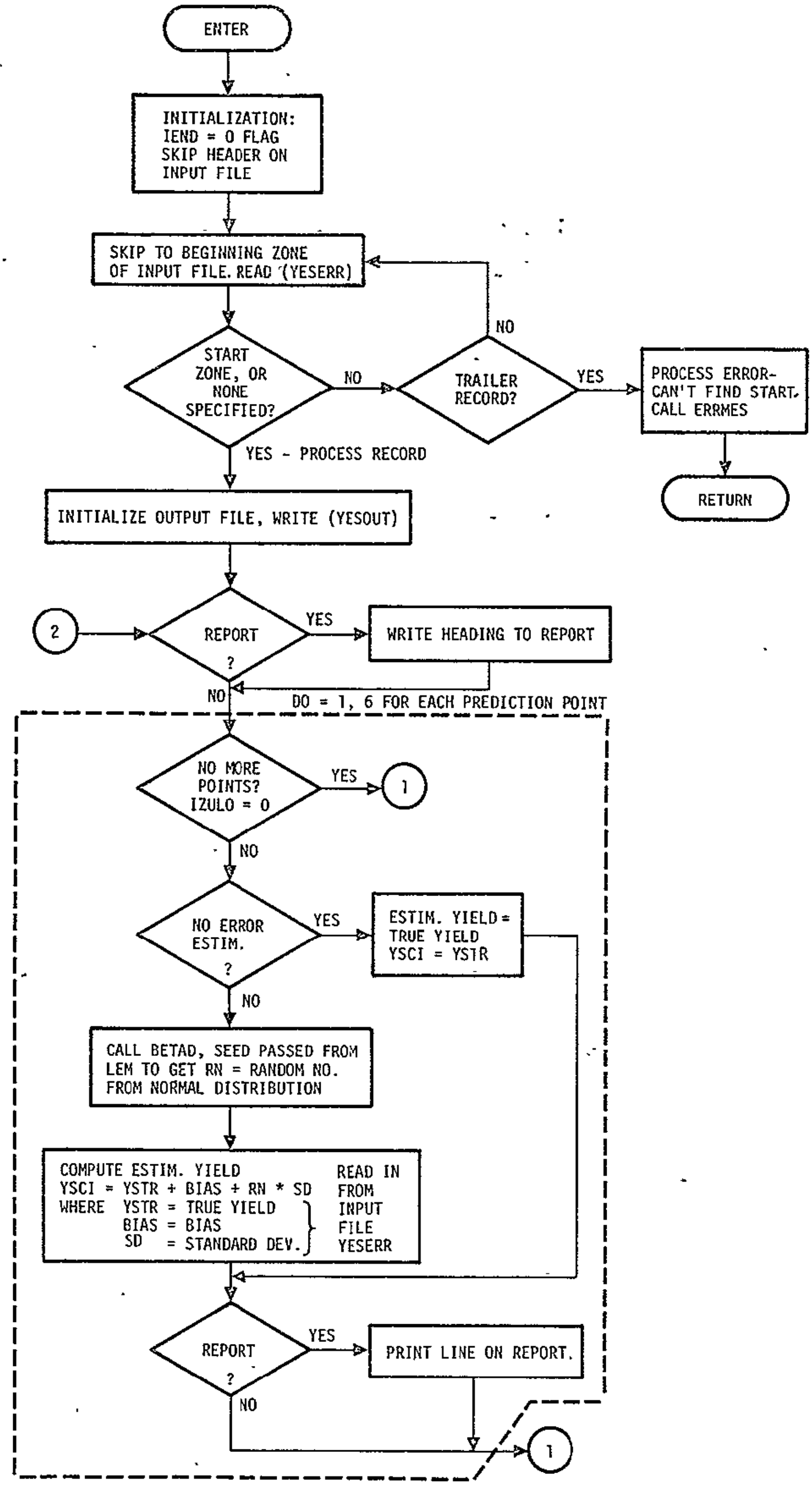
Special case:

If YSCI <0, YSCI = 0.0

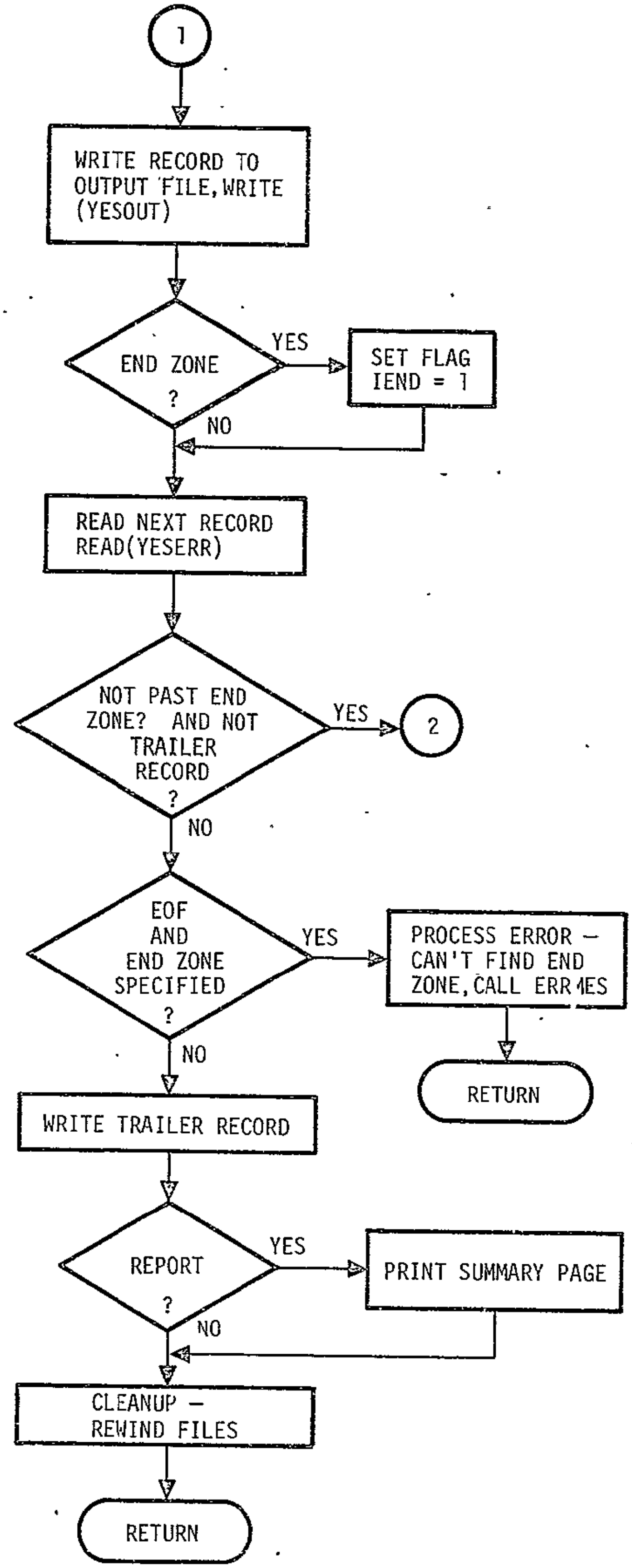
b. PERCNT =((ABS (YSCI - YSTR)) * 100)/(YSTR + ER)

where		
PERCNT	=	percent of error, always positive
YSCI	=	yield estimate
YSTR	=	true yield
ER	=	0.0 unless YSTR = 0, then ER = 10^{-6} to take care of this special case

YES FLOW DIAGRAM



YES FLOW DIAGRAM (CONT'D)



PART V

SUBROUTINE LISTINGS

EL ALGAMA,1,760430, 50102 , 1

```
      SUBROUTINE ALGAMA(X,GAMMA,IER)
C       CALCULATES THE GAMMA FUNCTION
C
C     X       INPUT
C     GAMMA   OUTPUT
C     IER     ERROR FLAG = 60 IF X NOT IN RANGE 0. TO 88.
C
      DIMENSION B(14),C(6)
       DOUBLE PRECISION B,C,F,G,HLNPI,Y,Z
      DATA B/-3.03019081028D-6,2.79832889938D-5,-1.21417348706320-4,    ALGAMA  *NEW
     *3.75365052263070-4,-8.37564685135170-4,2.00109185027554D-3,       ALGAMA  *NEW
     *-2.83646252937282D-3,1.11497143357789D-2,-2.66186594955060-4,     ALGAMA  *NEW
     *7.42489154194447D-2,8.15769261241555D-2,4.11840330166781D-1,      ALGAMA  *NEW
     *4.22784335510233SD-1,9.99999999999990D-1/                                 *NEW
      DATA C/-1.91752691752692D-3,8.41750841750842D-4,-5.95238095238095DALGAMA  *NEW
     *-4,7.93650793650794D-4,-2.77777777777778D-3,8.33333333333333D-2/  ALGAMA  *NEW
      DATA HLNPI/.918938533204673/                                              **-7
      GAMMA=0.
      IER=60
      IF (X .LE. 0.0 .OR. X .GT. 88.0) RETURN
      IER=0
      Y=X
      IF(Y.GE.10.) GO TO 60
      Z=DMOD(Y,1.0)
      IF(Z.NE.0.0) GO TO 20
      Z=1.0
      G=2.0
      GO TO 27
   20 G=B(1)
      DO 26 J=2,14
   26 G=G*Z+B(J)
   27 IF(Y-3.0) 28,45,38
   28 IF(Y.GT.2.0) GO TO 45
      G=G/Y
      IF(Y.LE.1.0) G=G/(Y+1.0)
      GO TO 45
   38 F=1.0
      K=DINT(Y-Z+.5)-2
      DO 40 J=1,K
      Y=Y-1.0
   40 F=F*Y
      G=G*F
   45 GAMMA=DLOG(G)
      RETURN
   60 Z=Y*Y
      G=C(1)
      DO 65 J=2,6
   65 G=G/Z+C(J)
      G=((G/Y+HLNPI)+((Y-.5)*DLOG(Y)-Y))
      GAMMA=G
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE APHDR                                                  APHDR
C           FILE DEFINITIONS AND RECORD LENGTHS                         FILFS
      COMMON  /FILES /                                                  FILFS
     1  SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACOUTS,LACO           FILFS
     2 ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO          FILFS
     3 ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILFS
     4 ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                       FILFS
      INTEGER  SEGID ,CROPW ,SUBHST,ACOUTS,CAMSF ,CAMERR,CASF  ,YESOUT  FILFS
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILFS
C                                                                       FILFS
C           PAGE EJECT CONTROL PARAMETERS FOR LEM                       PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1  NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                          PAGECM
C                                                                       PAGECM
C           LEM CONTROL CARD INPUT DATA                                 LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1  TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2 ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3 ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSFSG,ICSECM,ICSFSH,ICSECE  LEMCM
     4 ,ICSFYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6  LEMCM
     5 ,RSEED7,ICSFST,ICSECO,ICSEYS,ICSECU,ICSECD                       LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1 ,RSEED6,RSEED7                                                   LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C           FLAGS AND COUNTERS FOR CAS SIMULATOR                        CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1  H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS   CASFLG
     2 ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3 ,ENDC  ,ENDRLG,ENDZON,IRSTR ,IRZONE,IRREG                        CASFLG
     4 ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13   CASFLG
     5 ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR          CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C           STATISTICAL INFORMATION FOR LEM                             STATS
      COMMON  /STATS /                                                  STATS
     1  ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR               STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C           CAS CONTROL CARD INPUT DATA AND CONSTANTS                   CASCM
      COMMON  /CASCM /                                                  CASCM
     1  AREACF,YCF   ,PROCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)    CASCM
     2 ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)       CASCM
     3 ,WPRIOR(4)    ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
      CALL EJECT(14)                                                    APHDR
      WRITE (OUTP,1000)                                                 APHDR
 1000 FORMAT(/25X,72HA R E A     A N D     P R O D U C T I O N     S U M M APHDR
     1A R Y     R E P O R T //)                                         APHDR
      IF(PPFLG.EQ.0) WRITE(OUTP,2000) CUNTRY,IBW,NT                     APHDR
 2000 FORMAT(2X,7HCOUNTRY,2X,A6,7X,9HBIOWINDOW,5X,I1,15X,9HITERATION,   APHDR
     1 1X,I3/)                                                          APHDR
      IF(PPFLG.EQ.1) WRITE(OUTP,3000) CUNTRY,                           APHDR
     1        IPRD(2,IPD),IPRD(3,IPD),IPRD(1,IPD),NT                    APHDR
 3000 FORMAT(2X,7HCOUNTRY,2X,A6,7X,15HPREDICTION DATE ,2X,              APHDR
```

```
     1  2(I2,1H/),I2,5X,9HITERATION ,1X,I3/)                          APHDR
      WRITE(OUTP,4000)                                                 APHDR
 4000 FORMAT(15X,1H*,26X,8HA R E A ,25X,1H*,6X,10HY I E L D ,5X,      APHDR
     1  1H*,6X,20HP R O D U C T I O N  )                              APHDR
      INDX=AUNITS+1                                                    APHDR
      WRITE(OUTP,5000) (APRUTS(I,INDX),I=1,4),                         APHDR
     1    (YPRUTS(I,INDX),I=1,3),(PPRUTS(I,INDX),I=1,5)                APHDR
 5000 FORMAT(15X,1H*,20X,4A6,15X,1H*,2X,3A6,1X,1H*,3X,5A6/15X,1H*,59X, APHDR
     1  1H*,21X,1H*)                                                   APHDR
      WRITE(OUTP,6000)                                                 APHDR
 6000 FORMAT(2X,1HR,3X,1HZ,8X,1H*,23X,6HNO. IN,10X,3HNO.,3X,           APHDR
     1 7HCV AREA,2X,2HCV,3X,1H*,14X,8HST DEV *,18X,6HCV PRD,3X,2HCV)   APHDR
      WRITE(OUTP,7000)                                                 APHDR
 7000 FORMAT(2X,1HE,3X,1HO,8X,1H*,2X,,4HTRUE,5X,4HEST.,7X,             APHDR
     1  9HSUBSTRATA,5X, 8HSEGMENTS,2X,4HEST.,3X,5HERROR,1X,1H*,1X,     APHDR
     1  4HTRUE,3X,4HEST.,4X,6HPCT. *,2X,4HTRUE,5X,4HEST.,4X,4HEST.,3X, APHDR
     1    5HERROR)                                                     APHDR
      WRITE(OUTP,8000)                                                 APHDR
 8000 FORMAT(2X,1HG,3X,1HN,1X,8HSTRATA *,3X,2HWA,7X,2HWA,10X,5HGROUP,  APHDR
     1 7X,8HOF GROUP,2X,4HPCT.,3X,4HPCT.,2X,1H*,2(5HYIELD,2X),         APHDR
     1 6H ERROR,2H *,2X,4HPROD,5X,4HPROD,4X,2(4HPCT.,3X))              APHDR
      WRITE(OUTP,9000)                                                 APHDR
 9000 FORMAT(6X,1HE,8X,1H*,21X,1H1,4X,1H2,4X,1H3,5X,1H1,4X,1H2,3X,4HTRUEAPHDR
     1  ,3X,4HTRUE,2X,1H*,21X,1H*,19X,2(4HTRUE,3X)/)                   APHDR
      RETURN                                                           APHDR
      END                                                              APHDR
```

```
FOR,IS ASSCLS
      SUBROUTINE ASSCLS(IOPT)                                           ASSCLS
C     ASSIGNS CLASS NUMBERS TO ALL SUBSTRATA IN THE STRATA TABLES       ASSCLS
C        TABLES NECCESSARY TO DETERMINE CLASS SETS WITHIN A ZONE        CLSTAB
      COMMON /CLSTAB/                                                   CLSTAB
     1        ISTRAT(300),ISBSTR(300),NSCNT(300),IGROUP(300),IDAT1(300), MOD1
     2        IDAT2(300),XORD(300),IXPT(300),IRANK(300),IBPT(10),IEPT(10),MOD1
     3        MAXCLS,ICLCNT,ISUB1,NACQ                                  CLSTAB
      DIMENSION DAT1(300),DAT2(300),RANK(300)                           MOD1
      EQUIVALENCE (IDAT1(1),DAT1(1)),(IDAT2(1),DAT2(1)),(IRANK(1),      CLSTAB
     1RANK(1))                                                          CLSTAB
      DIMENSION ICT(10)                                                 ASSCLS
      DATA IDAT/-1000000/                                               ASSCLS
      DO 5 I=1,ISUB1                                                    ASSCLS
      IDAT2(I) = 0                                                      ASSCLS
    5 CONTINUE                                                          ASSCLS
      IF(IOPT - 1)25,20,10                                              ASSCLS
   10 DO 15 I=1,ISUB1                                                   ASSCLS
      IDAT2(I) = 1                                                      ASSCLS
   15 CONTINUE                                                          ASSCLS
   20 RETURN                                                            ASSCLS
   25 DO 50 I=1,ISUB1                                                   ASSCLS
      IF(NSCNT(I) .EQ. 0)GO TO 50                                       ASSCLS
      IPTR = IDAT1(I)                                                   ASSCLS
      DO 30 J=1,NACQ                                                    ASSCLS
      JPT = J                                                           ASSCLS
      IF(IPTR .EQ. IXPT(J))GO TO 35                                     ASSCLS
   30 CONTINUE                                                          ASSCLS
      GO TO 50                                                          ASSCLS
   35 DO 40 J=1,ICLCNT                                                  ASSCLS
      JCLS = J                                                          ASSCLS
      IF(JPT .GE. IBPT(J) .AND. JPT .LE. IEPT(J))GO TO 45               ASSCLS
   40 CONTINUE                                                          ASSCLS
      GO TO 50                                                          ASSCLS
   45 IDAT2(I) = JCLS                                                   ASSCLS
   50 CONTINUE                                                          ASSCLS
C    ASSIGN CLASS NUMBERS TO GROUP 2 SUBSTRATA WITH NO SEGMENTS         ASSCLS
      ISTRSV = ISTRAT(1)                                                ASSCLS
      I = 1                                                             ASSCLS
      IFLAG1 = 0                                                        ASSCLS
   55 IFLAG = 0                                                         ASSCLS
```

```
      ISTART = I                                                        ASSCLS
      DO 60 J=1,ICLCNT                                                  ASSCLS
      ICT(J) = 0                                                        ASSCLS
   60 CONTINUE                                                          ASSCLS
   65 IEND = I                                                          ASSCLS
      IF(IGROUP(I) .NE. 2 .OR. NSCNT(I) .EQ. 0)GO TO 70                 ASSCLS
      IFLAG = 1                                                         ASSCLS
      IP = IDAT2(I)                                                     ASSCLS
      ICT(IP) = ICT(IP) + 1                                             ASSCLS
   70 I = I + 1                                                         ASSCLS
      IF(I .LE. ISUB1)GO TO 75                                          ASSCLS
      IFLAG1 = 1                                                        ASSCLS
      GO TO 80                                                          ASSCLS
   75 IF(ISTRAT(I) .EQ. ISTRSV)GO TO 65                                 ASSCLS
      ISTRSV = ISTRAT(I)                                                ASSCLS
   80 IF(IFLAG .NE. 0)GO TO 85                                          ASSCLS
      IF(IFLAG1 .NE. 0)GO TO 20                                         ASSCLS
      GO TO 55                                                          ASSCLS
   85 IMAX = IDAT                                                       ASSCLS
      DO 90 J=1,ICLCNT                                                  ASSCLS
      IF(ICT(J).LE. IMAX)GO TO 90                                       ASSCLS
      IMAX = ICT(J)                                                     ASSCLS
      IJCL = J                                                          ASSCLS
   90 CONTINUE                                                          ASSCLS
      DO 95 J=ISTART,IEND                                               ASSCLS
      IF(IGROUP(J) .EQ. 2 .AND. IDAT2(J) .EQ. 0)IDAT2(J) = IJCL         ASSCLS
   95 CONTINUE                                                          ASSCLS
      IF(IFLAG1 .NE. 0)GO TO 20                                         ASSCLS
      GO TO 55                                                          ASSCLS
      END                                                               ASSCLS
```

```
      SUBROUTINE BETAD(SEED,XBAR,SIGMA,XI,IOPT,IER)
      DOUBLE PRECISION SEED
      INTEGER FLAG
      REAL K
      DATA P/88./
      DATA K/2./
      DATA FP/.00005/
      DATA SG /1.E-8 /
   50 CONTINUE
      FLAG=0
      I=0
      XAVG=XBAR
      XI=0.0
      IER=0
C
C  CHOOSE UNIFORM RANDOM NUMBER
      CALL RDM1A(SEED,P)
C
C COMPUTE T,CHK,RN, - NORMAL DISTRIB. PARAMETERS
      T=SQRT(ALOG(1.0/(P*P)))
      IF(P.GT.0.5)T=SQRT(ALOG(1.0/((1.0-P)*(1.0-P))))
      CHK=T-(2.30753+.27061*T)/(1.0+0.99229*T+.04481*T*T)
      RN=CHK
      IF(P.LE.0.5) RN=-RN
C
C  IF NORMAL DISTRIB. OPTION, JUMP OUT
      IF(IOPT.EQ.0) GO TO 70
      XI=RN
      RETURN
   70 CONTINUE
C
C  CHECK FOR END CASES
      IF(XBAR.LT.0. .OR.XBAR.GT.1.) IER=1
      IF(SIGMA.LT.0.0) IER=2
      IF (XBAR.GT.0.0.AND.XBAR.LT.1.0.AND.SIGMA.GT.0.0) GO TO 10
      XI=XBAR
      IF(XBAR.LT.0.)  XI=0.0
      IF(XBAR.GT.1.)  XI=1.
      RETURN
C
C SWITCH IF AVERAGE ABOVE HALF
   10 CONTINUE
      IF (XBAR.LE.0.5) GO TO 20
      FLAG=1
      XAVG=1.0-XBAR
C
C  COMPUTE UPPER LIMIT ON SIGMA
   20 CONTINUE
      SIGMAL=XAVG*SQRT((1.00-XAVG)/(XAVG+FP))
      SIG=SIGMA
      IF(SIGMA.LE.SIGMAL) GO TO 30
      SIG=SIGMAL
      IER=2
C
C  COMPUTE THE METHOD THRESHHOLD
   30 CONTINUE
      SIGT=XAVG*SQRT((1.0-XAVG)/(XAVG+K))
C
```

```
C   COMPUTE BETA PARAMETERS A AND B
      XSQ=XAVG*XAVG
      SIGSQ=SIG*SIG
      A=(XSQ-XAVG*(XSQ+SIGSQ))/SIGSQ
      B=((1.0-XAVG)/XAVG)*A
      IF(SIG.GT.SIGT) GO TO 40
C
C  APPROXIMATION METHOD
C
C
C  COMPUTE BETA APPROXIMATION PARAMETERS
      YP=-RN
      H=2.0/ (1.0/(2.0*A- 1.0) + 1.0/(2.0*B - 1.0))
      Y=(YP*YP-3.0)/6.0
      W=(YP*SQRT(H+Y))/
     1  H - (1.0/(2.0*B-1.0)-1.0/(2.0*A-1.0))*(Y+5.0/6.0-2.0/(3.0*H))
C
C  COMPUTE XI
      IF(ABS(ALOG(B)+2.*W).GT. 87.0) GO TO 50
      XI=A/(A+B*EXP(2.0*W))
      GO TO 60
C
C  ITERATIVE METHOD
C
C RECOMPUTE A AND B IF OVER LIMIT
   40 CONTINUE
      CHK=A+B
      IF(CHK.LT.R) GO TO 80
      BP=(B/CHK)*(R-1.)
      A=(BP/B)*A
      B=BP
   80 CONTINUE
      I=0
      PHI=1.
      PLO=0.
      XHI=1.
      XLO=0.
      XI=XAVG
  120 I=I+1
      IF(I.GT.35) GO TO 100
      CALL IBETAI(XI,A,B,PO,IER)
      IF(IER.GT.0) GO TO 100
      DIFF=ABS(PO-P)
      DIFF1=AMIN1(ABS(XI-XLO),ABS(XI-XHI))
      IF(DIFF.LE.EP.OR.DIFF1.LE.SG ) GO TO 60
C
C  CHECK IF DONE, WITHIN TOLERANCE
      IF(PO.LE.P) GO TO 105
      XHI=XI
      PHI=PO
      GO TO 106
  105 CONTINUE
      XLO=XI
      PLO=PO
  106 CONTINUE
      XI=(XHI+XLO)/2.
      GO TO 120
C ERROR RETURN
  100 CONTINUE
      IER=3
```

```
      XI=0.0
      RETURN
   60 CONTINUE
      IF(FLAG.EQ.1) XI=1.0 - XI
      RETURN
      END
```

```
FOR,IS BLKDTA
      BLOCK DATA                                                            BLKDTA
C         BLOCK DATA ROUTINE FOR THE LEM PROGRAM                            BLKDTA
C                                                                           BLKDTA
C         COMMON BLOCK DEFINITIONS                                          BLKDTA
C         ARGUMENT LIST FOR ERROR PROCESSING                                ARGLST
      COMMON  /ARGLST/                                                      ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                                ARGLST
      DIMENSION  IARG(10)                                                   ARGLST
      EQUIVALENCE  ( IARG,ARG )                                             ARGLST
C                                                                           ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                         CASCM
      COMMON  /CASCM /                                                      CASCM
     1   AREACF,YCF   ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)       CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)          CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)               CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE            CASCM
C                                                                           CASCM
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                              CASFLG
      COMMON  /CASFLG/                                                      CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS      CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS            CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                           CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13      CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR            CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                      CASFLG
C                                                                           CASFLG
C         CONSTANT QUANTITIES FOR LEM PROGRAM                               CONST
      COMMON  /CONST /                                                      CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                           CONST
C                                                                           CONST
C         FILE DEFINITIONS AND RECORD LENGTHS                               FILES
      COMMON  /FILES /                                                      FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ              FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO             FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD             FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                          FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT      FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                    FILES
C                                                                           FILES
C         INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                       IXCASF
```

```
      COMMON  /IXCASF/                                                  IXCASF
     1  IXCASF(1),LIXCAS
C                                                                       IXCASF
C         INDEX RECORD FOR CAS DISTRIBUTION FILE                        IXDISF
      COMMON  /IXDISF/                                                  IXDISF
     1   IXDISF(1),LIXDIS
C        NOTE... 506 ONLY ALLOWS UP TO 8 PREDICTION POINTS INCLUDING    IXDISF
C        BIOWINDOWS ( 506 = 1 + 1 + 8*63, INDEX + HEADER + 8 PRED. PTS.)IXDISF
C                                                                       IXDISF
      COMMON/FILES1/                                                    FILES1
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C         INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE  IXSUBH
      COMMON  /IXSUBH/                                                  IXSUBH
     1 LIXSSH,IXSUBH(1)                                                 MOD1
C                                                                       IXSUBH
C         PAGE EJECT CONTROL PARAMETERS FOR LEM                         PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C         STATISTICAL INFORMATION FOR LEM                               STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C                                                                       BLKDTA
C *********************************************************************BLKDTA
C                                                                       BLKDTA
C                                                                       BLKDTA
      DATA  NERRS ,NFATAL,NPERRS,NARG                                   BLKDTA
     1   /    0  ,    0  ,    0  ,  0  /                                BLKDTA
C                                                                       BLKDTA
      DATA  APRUTS  / 6H(TEN T,6HHOUSAN,6HD ACRE,2HS)                   BLKDTA
     1              ,6H(THOUS,6HAND HE,6HCTARES,1H)  /                  BLKDTA
      DATA  PPRUTS  / 6H(HUNDR,6HED THO,6HUSAND ,6HBUSHEL,2HS)          BLKDTA
     1              ,6H(THOUS,6HAND ME,6HTRIC T,6HONS)  ,1H   /         BLKDTA
      DATA  YPRUTS  / 6H (BUSH,6HELS/AC,3HRE)                           BLKDTA
     1              ,6H(QUINT,6HALS/HE,6HCTARE)  /                      BLKDTA
C                                                                       BLKDTA
      DATA  AREAPS  / 10289.712 /                                       BLKDTA
C                                                                       BLKDTA
      DATA  ENDFIL,NTRMX ,MAXR  ,MAXZ  ,IMXSEG                          BLKDTA
```

```
     1    / 4HZZZZ, 100   , 999   , 999   , 150   /                       BLKDTA
C                                                                         JULY76
      DATA  MXCLSS  / 10 /                                                JULY76
C                                                                         BLKDTA
      DATA  SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ         BLKDTA
     1    /    1   ,  17  ,   2  ,  33  ,  3   , 168  ,  12  , 107  /    BLKDTA
      DATA  CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO        BLKDTA
     1    /    7   ,  19  ,   8  ,  50  , 14   , 504  ,  10  ,  23  /    BLKDTA
      DATA  SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD        BLKDTA
     1    /    9   ,  59  ,  11  ,  23  ,  13  ,  16  ,   4  , 303  /    BLKDTA
      DATA  INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                     BLKDTA
     1    /  5    ,  6   , 16   , 129  ,  15  ,  81    /                 JULY76
      DATA  ISUBH2,LSUBH2                                                 JULY76
     1    /   17  ,  39  /                                                JULY76
C                                                                         BLKDTA
      DATA  MXLINE  / 40 /                                                BLKDTA
C                                                                         BLKDTA
      DATA  ITER  ,NSEGTR,NCAMSR,NYESR ,NCASCR,NCASDR                     BLKDTA
     1    /   0   ,   0  ,   0  ,  0   ,   0  ,   0    /                 BLKDTA
C                                                                         BLKDTA
      DATA  LIXCAS,LIXDIS                                                 BLKDTA
     1    /   388 , 506  /                                                BLKDTA
C        TEMPORARILY LIXDIS = 506 ALLOWING UP TO 8 PREDICTION POINTS.     BLKDTA
C                                                                         BLKDTA
      DATA  LIXSSH  /3200 /                                               JULY76
C        TEMPORARILY LIXSSH = 200 ALLOWING UP TO 200 SUBSTRATA PER        JULY76
C        COUNTRY                                                          JULY76
C                                                                         JULY76
      DATA  LDS1 , LDS4 , LDS7 , LDS8 , LDS9 , LDS10, LDS11, LDS12        JULY76
     1    /  12  ,  24  ,  81  ,  10  ,   9  ,  20  ,  19  ,  19  /       JULY76
      DATA  LDS13, LDS14, LDS15, LDS16, LDS17                             BLKDTA
     1    /  25  ,  22  ,  22  ,  22  ,  28  /                            BLKDTA
C                                                                         BLKDTA
      END                                                                 BLKDTA
```

```
@ ELT CAMERS,1,760427, 39104    , 1

      SUBROUTINE CAMERS(INO)                                            CAMERS
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHSI,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO        FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHSI,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
      GO TO (10,20,30,40,50,60,70),INO                                  CAMERS
   10 CONTINUE                                                          CAMERS
      WRITE(OUTP,1000)IARG(1),IARG(2)                                   CAMERS
      RETURN                                                            CAMERS
   20 CONTINUE                                                          CAMERS
      WRITE(OUTP,2000)IARG(1),IARG(2)                                   CAMERS
      RETURN                                                            CAMERS
   30 CONTINUE                                                          CAMERS
      WRITE(OUTP,3000)IARG(1),IARG(2)                                   CAMERS
      RETURN                                                            CAMERS
   40 CONTINUE                                                          CAMERS
      WRITE(OUTP,4000)IARG(1),IARG(2)                                   CAMERS
      RETURN                                                            CAMERS
   50 CONTINUE                                                          CAMERS
      WRITE(OUTP,5000)IARG(1), ARG(2)                                   CAMERS
      RETURN                                                            CAMERS
   60 CONTINUE                                                          CAMERS
      WRITE(OUTP,6000)IARG(1),IARG(2), ARG(3)                           CAMERS
      RETURN                                                            CAMERS
   70 CONTINUE                                                          CAMERS
      WRITE(OUTP,7000)IARG(1),IARG(2),IARG(3)                           CAMERS
      RETURN                                                            CAMERS
 1000 FORMAT(2X,5HCAMS ,I2,20H MODEL NOT 1 OR 2 -  ,I1)                 CAMERS
 2000 FORMAT(2X,5HCAMS ,I2,30H ITMAX NOT BETWEEN 0 AND 99 -  ,I2)       CAMERS
 3000 FORMAT(2X,5HCAMS ,I2,29H IWIND NOT BETWEEN 0 AND 4 -  ,I1)        CAMERS
 4000 FORMAT(2X,29HBAD CAMS ID OR SEQUENCE NO. -  ,A4,1X,I2)            CAMERS
 5000 FORMAT(2X,5HCAMS ,I2,36H CROP CALENDAR COFF. OUT OF RANGE -  ,    CAMERS
     1  F7.3)                                                           CAMERS
 6000 FORMAT(2X,5HCAMS ,I2,35H BAD MULTI-TEMPORAL MATRIX VALUE M(,I1,   CAMERS
     1   4H) - ,F5.4)                                                   CAMERS
 7000 FORMAT(2X,5HCAMS ,I2,                                             CAMERS
     1   40H BAD MULTI-TEMPORAL MATRIX VALUE IGROUP(  ,I2,4H) - ,I1)    CAMERS
      END                                                               CAMERS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE CAMER2(INO)                                            CAMER2
C                                                                       CAMER2
C  THIS SUBROUTINE CONTAINS THE PROCESSING ERROR MESSAGES FOR THE       CAMER2
C     CAMS MODULE                                                       CAMER2
C                                                                       CAMER2
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
      GO TO (10,20,30,40),INO                                           CAMER2
   10 CONTINUE                                                          CAMER2
      WRITE(OUTP,1000)                                                  CAMER2
      RETURN                                                            CAMER2
   20 CONTINUE                                                          CAMER2
      WRITE(OUTP,2000)                                                  CAMER2
      RETURN                                                            CAMER2
   30 CONTINUE                                                          CAMER2
      WRITE(OUTP,3000)IARG(1)                                           CAMER2
      RETURN                                                            CAMER2
   40 CONTINUE                                                          CAMER2
      WRITE(OUTP,4000)IARG(1)                                           CAMER2
      RETURN                                                            CAMER2
 1000 FORMAT(2X,                                                        CAMER2
     160HCAMS INPUT TAPE SEGTRU - BEGINNING REGION AND ZONE NOT FOUND  )CAMER2
 2000 FORMAT(2X,                                                        CAMER2
     1  57HCAMS INPUT TAPE SEGTRU - ENDING REGION AND ZONE NOT FOUND   )CAMER2
 3000 FORMAT(2X,                                                        CAMER2
     1  16HCAMS INPUT TAPE  ,A6,17H - MISSING RECORD  )                 CAMER2
 4000 FORMAT(2X,33HDELTA DISTRIBUTION ERROR - FLAG =  ,I1)              CAMER2
      END                                                               CAMER2
```

```
      SUBROUTINE CAMS                                                   CAMS
C                                                                       CAMS
C    THIS SUBROUTINE IS THE DRIVER FOR THE CAMS MODULE, WHICH CALCULATESCAMS
C       THE ESTIMATED PROPORTION OF WHEAT.                              CAMS
C                                                                       CAMS
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,            TRAINS
     1   TTWIN(4,25),TTTOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),         TRAINS
     1   TPTRUE,TIZULU(4),TPEST(4),TPERR(4),TERTOT(3),TM(3),TV(3),TB(3) TRAINS
      INTEGER TIZULU                                                    TRAINS
      DIMENSION ITRAIN(129)                                             TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                         TRAINS
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER   ,NSEGTR,NCAMSL,NYESR ,NREC(7),NCASCR,NCASDR             STATS
      EQUIVALENCE  ( NI,ITER )                                          STATS
C                                                                       STATS
      COMMON/ERROR/TITL(4),IDATE,PESTIM,TOT,ALOCAL,ERTOT(3)             ERROR
     1   ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,       ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                         ERROR
      DIMENSION IERS(40)                                                ERROR
      EQUIVALENCE(TITL,IERS)                                            ERROR
      REAL MULT                                                         ERROR
      INTEGER TID,CROPD                                                 ERROR
C        CAMS CONTROL CARD INPUT DATA                                   CAMSCM
      COMMON/CAMSCM/ IMODEL,INULTI,ISIGFX,ISKIP,ITMAX,IREP,IWIND,       CAMSCM
     1    IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                    CAMSCM
      REAL MS                                                           CAMSCM
C                                                                       CAMSCM
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASF ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACO  ,ICLASS,ISEXI ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECU,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,RSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGFXT,LSIGFX,YESERR,LYESER,SEGIPU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,IACQ  ,LIACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
```

```
      1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF                 FILES
C                                                                           FILES
C        ARGUMENT LIST FOR ERROR PROCESSING                                 ARGLST
      COMMON  /ARGLST/                                                      ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                               ARGLST
      DIMENSION  IARG(10)                                                   ARGLST
      EQUIVALENCE   ( IARG,ARG )                                            ARGLST
C                                                                           ARGLST
      COMMON/SEGTRU/COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,                  SEGTRU
     1   IT,IPRIOR(6),ISPW,PT(2)                                            SEGTRU
      COMMON/CAMERR/ COUN2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,                 CAMERR
     1  PW(3,4),BERR(3,4),SIGERR(3,4)                                       CAMERR
      COMMON/ACQUIS/COUN1,IREG1,IZONE1,ISTRA1,ISUB1,ISEG1,                  ACQUIS
     1    IWIN(4,25),ITOTAL                                                 ACQUIS
      COMMON/CROPW/COUN3,IREG3,IZONE3,ISTRA3,ISUB3,                         CROPW
     1  START(2,4),END(2,4),SD(2),ERR(2,5)                                  CROPW
      INTEGER START,END,SD,ERR                                              CROPW
      COMMON/SIGEX/COUN5,IREG5,IZONE5,ZB(3,2),ZSIG(3,2,6)                   SIGEX
      COMMON/CAMSF/COUN6,IREG6,IZONE6,ISTRA6,ISUB6,ISEG6,                   CAMSF
     1    PIPUF,IZULU(4),PEST(4),PERR(4)                                    CAMSF
      COMMON/INDX/ INDEX(    1),IPOINT(2001),IPNT2(2001),IPEND,IPIN         INDX
      INTEGER WHE, MIX,OTH,SEASON,WINDOW,TYPE,MODEL                         CAMS
      REAL M                                                                CAMS
      DIMENSION HEAD(4,4),XI(3),M(3),BCC(3),SIGCC(3),P(3),IWINDO(4)         CAMS
      DIMENSION HD(16)                                                      CAMS
      EQUIVALENCE(HEAD,HD)                                                  CAMS
      DATA HD/ 4H**** ,4HWIND   ,4HOW 1   ,4H****   ,                       CAMS
     1         4H**** ,4HWIND   ,4HOW 2   ,4H****   ,                       CAMS
     1         4H**** ,4HWIND   ,4HOW 3   ,4H****   ,                       CAMS
     1         4H**** ,4HWIND   ,4HOW 4   ,4H****   /                       CAMS
      DATA WHE, MIX,OTH/1,2,3/,IFILL/0/
      DATA ZZZZ/4HZZZZ/                                                     CAMS
      IPEND = 0
      IPIN = 0
      IMP = 4
      IEND = 0
      IUSE = 0
      NSTTL=48                                                              CAMS
      SUBTTL(1)=1H                                                          CAMS
      SUBTTL(2)=1H                                                          CAMS
      SUBTTL(3)=1H                                                          CAMS
      SUBTTL(4)=6HCAMS P                                                    CAMS
      SUBTTL(5)=6HROPORT                                                    CAMS
      SUBTTL(6)=6HION ES                                                    CAMS
      SUBTTL(7)=6HTIMATE                                                    CAMS
      SUBTTL(8)=6H DATA                                                     CAMS
      SUBTTL(9)=6HREPORT                                                    CAMS
      SUBTTL(10) = 6H
      IF(PRINTF.LE.0) GO TO 5                                               CAMS
      CALL EJECT(1)                                                         CAMS
      WRITE(OUTP,1001)                                                      CAMS
 1001 FORMAT(2X)                                                            CAMS
    5 CONTINUE                                                              CAMS
      NREC(2)=0                                                             CAMS
      NREC(4)=0                                                             CAMS
      NREC(6)=0                                                             CAMS
      NREC(7)=0                                                             CAMS
      NCAMSP=0                                                              CAMS
      COUN4=4H                                                                        *NEW
      IREG4=0                                                                         *NEW
```

```
      IZONE4=0                                                          *NEW
      ISTRA4=0                                                          *NEW
      ISUB4=0                                                           *NEW
      ISEG4=0                                                           *NEW
      COUN3=4H                                                          *NEW
      IREG3=0                                                           *NEW
      IZONE3=0                                                          *NEW
      ISTRA3=0                                                          *NEW
      ISUB3=0                                                           *NEW
      COUN5=4H                                                          *NEW
      IREG5=0                                                           *NEW
      IZONE5=0                                                          *NEW
C                                                                   CAMS
C  INITIALIZATION                                                   CAMS
C       SET FILE FLAGS                                              CAMS
      ISEG=0                                                        CAMS
      ICROPW=0                                                      CAMS
      IF(IACQ.EQ.0.AND.(ICAMS.EQ.3.OR.ICLASS.EQ.2.OR.ISCC.EQ.2))    CAMS
     1  ICROPW=1                                                    CAMS
      ISIG=0                                                        CAMS
      IF(IACQ.EQ.1.OR.ICAMS.EQ.3.OR.ISEXT.EQ.2) ISIG=1              CAMS
      CALL INITI(ISEG,IACQ,IACQ,ICROPW,ISIG,HEAD,ITSEG)             CAMS
C                                                                   CAMS
C  CHECK FOR ERROR                                                  CAMS
      IF(NFATAL.GT.0)RETURN                                         CAMS
      IF(IACQ.EQ.0) GO TO 1000                                      CAMS
C                                                                   CAMS
C  SPECIAL CASE - NO ACQUIS. FILE                                   CAMS
   10 CONTINUE                                                      CAMS
C  ZERO OUT ERROR PARAMETERS                                        CAMS
      DO 12 I=7,40                                                  CAMS
      IERS(I)=0                                                     CAMS
   12 CONTINUE                                                      CAMS
   11 CONTINUE                                                      CAMS
      CALL INPT(0,1,1,0,1,0,IDONE,IEND)                             CAMS
      IF(IDONE.GT.0) GO TO 4000                                     CAMS
C                                                                   CAMS
C  SET UP OUTPUT FILE RECORD                                        CAMS
      COUN6=COUN4                                                   CAMS
      IREG6=IREG4                                                   CAMS
      IZONE6=IZONE4                                                 CAMS
      ISTRA6=ISTRA4                                                 CAMS
      ISUB6=ISUB4                                                   CAMS
      ISEG6=ISEG4                                                   CAMS
      PTRUE=PT(VHF)                                                 CAMS
      SEASON=ISPW+1                                                 CAMS
      DO 30 WINDOW=1,4                                              CAMS
      IZULU(WINDOW)=START(SEASON,WINDOW)                            CAMS
      PEST(WINDOW)=PTRUE                                            CAMS
      PERR(WINDOW)=0.                                               CAMS
   30 CONTINUE                                                      CAMS
C                                                                   CAMS
C  IF REPORT, DO REPORT                                             CAMS
      IF(PRINTF.LE.0) GO TO 40                                      CAMS
      DO 60 WINDOW=1,4                                              CAMS
      DO 52 I=1,4                                                   CAMS
      TITL(I)=HEAD(I,WINDOW)                                        CAMS
   52 CONTINUE                                                      CAMS
      IDATE=IZULU(WINDOW)                                           CAMS
      PESTIM=PTRUE                                                  CAMS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL REPORT(0,WINDOW,0)                                           CAMS
   60 CONTINUE                                                          CAMS
C                                                                       CAMS
C   OUTPUT RECORD                                                       CAMS
   40 CONTINUE                                                          CAMS
      WRITE(CAMSF)COUN6,IREG6,IZONE6,ISTRA6,ISUB6,ISEG6,PTRUE,          CAMS
     1   (IZULU(WINDOW),PEST(WINDOW),PERR(WINDOW),WINDOW=1,4)           CAMS
      NCAMSR=NCAMSR+1                                                   CAMS
      GO TO 11                                                          CAMS
C***********************************************************************CAMS
C                                                                       CAMS
C   PASS 1 - TRAINING SEGMENTS                                          CAMS
 1000 CONTINUE                                                          CAMS
      CALL INPT(ISEG,IACQ,IACQ,ICROPW,1,1,IDONE,IEND)                   CAMS
      IF(IDONE.GT.0) GO TO 1900                                         CAMS
C                                                                       CAMS
C   INITIALIZATION                                                      CAMS
      MODEL=3                                                           CAMS
      IF(IMODEL.EQ.2)MODEL=1                                            CAMS
      SEASON=ISPW+1                                                     CAMS
      IFIRST=0                                                          CAMS
      DO 20 I=1,4                                                       CAMS
      IWINDO(I)=0                                                       CAMS
   20 CONTINUE                                                          CAMS
C                                                                       CAMS
C   ZERO OUT IACQ RECORD (CAMSF PART)                                   CAMS
      DO 50 WINDOW=1,4                                                  CAMS
      IIZULU(WINDOW)=0                                                  CAMS
      TPEST(WINDOW)=0.                                                  CAMS
      TPERR(WINDOW)=0.                                                  CAMS
   50 CONTINUE                                                          CAMS
      WINDOW=0                                                          CAMS
  100 CONTINUE                                                          CAMS
      IACQNO=0                                                          CAMS
      WINDOW=WINDOW+1                                                   CAMS
      IF(WINDOW.GT.4) GO TO 101                                         CAMS
  200 CONTINUE                                                          CAMS
      IACQNO=IACQNO+1                                                   CAMS
      IF(IACQNO.GT.25) GO TO 100                                        CAMS
C                                                                       CAMS
C   GET NEXT ACQUISITION, JUMP OUT IF NO MORE                           CAMS
      IF(IWIN(WINDOW,IACQNO).EQ.0) GO TO 100                            CAMS
      IF(IACQNO.EQ.1) IWINDO(WINDOW)=1                                  CAMS
C                                                                       CAMS
C   ZERO OUT ERROR VALUES                                               CAMS
      DO 240 I=7,40                                                     CAMS
      IERS(I)=0                                                         CAMS
  240 CONTINUE                                                          CAMS
      DO 220 I=1,4                                                      CAMS
      IERS(I)=4H                                                        CAMS
  220 CONTINUE                                                          CAMS
C                                                                       CAMS
C   ENTRY POINT FOR  SPECIAL CASE - ORDINARY SEGMENT WITH NO CORREL.    CAMS
C      GET CROPW RECORD - CALL INPT - IF NECESSARY                      CAMS
 3000 CONTINUE                                                          CAMS
      IF(IUSE .GT. 6)CALL INPT(1,1,1,ICROPW,1,0,IDONE,IEND)
      IF(IDONE.GT.0) GO TO 4000                                         CAMS
C                                                                       CAMS
C   SET BYPASS VALUES FOR CLASSIFICATION  ERROR                         CAMS
      DO 300 TYPE=1,3                                                   CAMS
```

```
      MULT(TYPE)=1.                                                CAMS
      XI(TYPE)=PW(TYPE,WINDOW)                                     CAMS
  300 CONTINUE                                                     CAMS
C                                                                  CAMS
C   IF BYPASS CLASSIF. ERROR, SKIP DOWN                            CAMS
      IF(ICLASS.EQ.2.OR.ICAMS.EQ.3) GO TO 700                      CAMS
C                                                                  CAMS
C   SET BYPASS VALUES FOR MULTI-TEMPORAL ERROR                     CAMS
      DO 410 TYPE=1,MODEL                                          CAMS
      M(TYPE)=1.                                                   CAMS
  410 CONTINUE                                                     CAMS
C                                                                  CAMS
C   IF BYPASS MULTI-TEMP.ERROR, SKIP DOWN                          CAMS
      IF(IMULTI.GE.1) GO TO 500                                    CAMS
      DO 420 TYPE=1,MODEL                                          CAMS
      CALL MULTI(TYPE,SEASON,IWINDO,M(TYPE))                       CAMS
  420 CONTINUE                                                     CAMS
C                                                                  CAMS
C   SET BYPASS VALUES FOR CROP CALENDAR ERROR                      CAMS
  500 CONTINUE                                                     CAMS
      DO 510 TYPE=1,MODEL                                          CAMS
      BCC(TYPE)=0.                                                 CAMS
      SIGCC(TYPE)=0.                                               CAMS
  510 CONTINUE                                                     CAMS
C                                                                  CAMS
C   IF BYPASS CROP CAL. ERROR, SKIP DOWN                           CAMS
      IF(ISCC.GE.2) GO TO 600                                      CAMS
      DO 520 TYPE=1,MODEL                                          CAMS
      CALL CROP(SEED(4),TYPE,SEASON,WINDOW,IACQNO,                 CAMS
     1     BCC(TYPE),SIGCC(TYPE),ITSEG)                            CAMS
  520 CONTINUE                                                     CAMS
C                                                                  CAMS
C   COMPUTE CLASSIFICATION ERROR                                   CAMS
  600 CONTINUE                                                     CAMS
      XI(MIX)=0.                                                   CAMS
      XI(OTH)=0.                                                   CAMS
      DO 620 TYPE=1,MODEL                                          CAMS
      CALL CLASS(SEED(2),TYPE,WINDOW,M(TYPE),BCC(TYPE),SIGCC(TYPE),  CAMS
     1  XI(TYPE))                                                  CAMS
  620 CONTINUE                                                     CAMS
C                                                                  CAMS
C   COMPUTE PEST                                                   CAMS
  700 CONTINUE                                                     CAMS
      P(WHE)=100.                                                  CAMS
      P(MIX)=0.                                                    CAMS
      P(OTH)=0.                                                    CAMS
      IF(IMODEL.EQ.2) GO TO 710                                    CAMS
      P(WHE)=PI(WHE)-PI(MIX)*PW(MIX,IWIND)                         CAMS
      P(MIX)=PI(MIX)                                               CAMS
      P(OTH)=100.-P(MIX)-P(WHE)                                    CAMS
  710 CONTINUE                                                     CAMS
      PES=P(WHE)*XI(WHE)+P(MIX)*XI(MIX)+P(OTH)*XI(OTH)             CAMS
C                                                                  CAMS
C   EXIT POINT FOR SPECIAL CASE - ORDINARY SEGMENT WITH NO CORREL.  CAMS
      IF(TUSE.GT.6) GO TO 2750                                     CAMS
      DO 930 TYPE=1,3                                              CAMS
      TMM(TYPE,WINDOW,IACQNO)=MULT(TYPE)                           CAMS
      TBB(TYPE,WINDOW,IACQNO)=CLBIAS(TYPE)                         CAMS
      TVV(TYPE,WINDOW,IACQNO)=CLRAND(TYPE)                         CAMS
  930 CONTINUE                                                     CAMS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                                       CAMS
C   IF REPORT, PRINT REPORT                                             CAMS
      IF(PRINTF.LE.0) GO TO 900                                         CAMS
      IF(IACQNO.GT.1) GO TO 810                                         CAMS
      DO 820 I=1,4                                                      CAMS
      TITL(I)=HEAD(I,WINDOW)                                            CAMS
  820 CONTINUE                                                          CAMS
  810 CONTINUE                                                          CAMS
      IDATE=IWIN(WINDOW,IACQNO)                                         CAMS
      PESTIM=PES                                                        CAMS
      TOT=PES-PT(WHE)                                                   CAMS
      IFIRST=IFIRST+1                                                   CAMS
      CALL REPORT(1,IFIRST,IREP)                                        CAMS
C                                                                       CAMS
C   STORE VALUES IN OUTPUT RECORD                                       CAMS
  900 CONTINUE                                                          CAMS
      IF(IACQNO.GT.1) GO TO 200                                         CAMS
      TIZULU(WINDOW)=IWIN(WINDOW,IACQNO)                                CAMS
      TPEST(WINDOW)=PES                                                 CAMS
      TPERR(WINDOW)=PES-PT(WHE)                                         CAMS
      GO TO 200                                                         CAMS
  101 CONTINUE                                                          CAMS
C                                                                       CAMS
C   SAVE ON SCRATCH DA FILE TACQ                                        CAMS
      COUN7=COUN4                                                       CAMS
      IREG7=IREG4                                                       CAMS
      IZONE7=IZONE4                                                     CAMS
      ISTRA7=ISTRA4                                                     CAMS
      ISUB7=ISUB4                                                       CAMS
      ISEG7=ISEG4                                                       CAMS
      TPTRUE=PT(WHE)                                                    CAMS
      IF(ITSEG.GT.0) GO TO 915                                          CAMS
      WRITE(CAMSI)COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,TPTRUE,         CAMS
     1  (TIZULU(I),TPEST(I),TPERR(I),I=1,4)                             CAMS
      NCAMSR= NCAMSR + 1                                                CAMS
      GO TO 1000                                                        CAMS
  915 CONTINUE                                                          CAMS
      DO 910 I=1,4                                                      CAMS
      DO 920 J=1,25                                                     CAMS
      ITWIN(I,J)=IWIN(I,J)                                              CAMS
  920 CONTINUE                                                          CAMS
  910 CONTINUE                                                          CAMS
      ITTOT=ITOTAL                                                      CAMS
      CALL TSAVE(ISEG7,2,IBAD)                                          CAMS
      IF(NFATAL.GT.0) RETURN                                            CAMS
      GO TO 1000                                                        CAMS
 1900 CONTINUE                                                          CAMS
      IF(IUOFF.GE.2) RETURN                                             CAMS
      IF(ITSEG.EQ.0) GO TO 4000                                         CAMS
C***********************************************************************CAMS
C                                                                       CAMS
C   PASS 2 - ORDINARY SEGMENTS                                          CAMS
 2000 CONTINUE                                                          CAMS
C                                                                       CAMS
C FINISH LAST WRITE TO SCRATCH FILE TACQ                                CAMS
      CALL TSAVE(0,3,IBAD)                                              CAMS
C                                                                       CAMS
C   REINITIALIZE FILES                                                  CAMS
      CALL INITI(ISEG,IACQ,TACQ,ICROPW,ISIG,HEAD,ITSEG)                 CAMS
      IACQNO=1                                                          CAMS
```

```
C                                                                       CAMS
 2010 CONTINUE                                                          CAMS
      CALL INPT(ISEG,IACQ,IACQ,1,ISIG,2,IDONE,IEND)                     CAMS
      IF(IDONE.GT.0) GO TO 4000                                         CAMS
      IFIRST=0                                                          CAMS
      SEASON=ISPW+1                                                     CAMS
      DO 2005 I=1,4                                                     CAMS
      IWINDO(I)=0                                                       CAMS
 2005 CONTINUE                                                          CAMS
C                                                                       CAMS
C   ZERO OUT CAMSE RECORD                                               CAMS
      DO 2020 WINDOW=1,4                                                CAMS
      IZOLU(WINDOW)=0.                                                  CAMS
      PEST(WINDOW)=0.                                                   CAMS
      PERR(WINDOW)=0.                                                   CAMS
 2020 CONTINUE                                                          CAMS
      WINDOW=0                                                          CAMS
 2100 CONTINUE                                                          CAMS
      WINDOW=WINDOW+1                                                   CAMS
      IF(WINDOW.GT.4) GO TO 2101                                        CAMS
      IF(IWIN(WINDOW,1).EQ.0) GO TO 2100                                CAMS
      IWINDO(WINDOW)=1                                                  CAMS
C                                                                       CAMS
C   ZERO OUT ERROR VALUES                                               CAMS
      DO 2110 I=7,40                                                    CAMS
      IERS(I)=0                                                         CAMS
 2110 CONTINUE                                                          CAMS
      DO 2120 I=1,4                                                     CAMS
      IERS(I)=4H                                                        CAMS
 2120 CONTINUE                                                          CAMS
C                                                                       CAMS
C   CORRELATE WITH TRAINING SEGMENT                                     CAMS
      CALL CORREL(ITMAX,IWIN(WINDOW,1),WINDOW,IUSE)                     CAMS
      IF(IUSE.GT.6.AND.ISKIP.LE.0) GO TO 2100                           CAMS
      IF(IUSE.GT.6.AND.ISKIP.GE.1) GO TO 3000                           CAMS
      DO 2200 TYPE=1,3                                                  CAMS
      XI(TYPE)=0.                                                       CAMS
 2200 CONTINUE                                                          CAMS
C                                                                       CAMS
C   CALCULATE SIGNATURE EXTENSION ERROR                                 CAMS
      IF(ISEXT.EQ.2.OR.ICAMS.EQ.3) GO TO 2400                           CAMS
      DO 2500 TYPE=1,MODEL                                              CAMS
      CALL SGEXT(SEED(3),TYPE,WINDOW,IUSE,XI(TYPE))                     CAMS
 2500 CONTINUE                                                          CAMS
      GO TO 2600                                                        CAMS
C                                                                       CAMS
C   SET BYPASS VALUES FOR SIGNATURE EXTENSION                           CAMS
 2400 CONTINUE                                                          CAMS
      DO 2300 TYPE=1,MODEL                                              CAMS
      XI(TYPE)=PW(TYPE,WINDOW)*(1.+TERTOT(TYPE))                        CAMS
 2300 CONTINUE                                                          CAMS
C                                                                       CAMS
C   COMPUTE PEST                                                        CAMS
 2600 CONTINUE                                                          CAMS
      P(WHE)=100.                                                       CAMS
      P(MIX)=0.                                                         CAMS
      P(OTH)=0.                                                         CAMS
      IF(IMODEL.EQ.2) GO TO 2610                                        CAMS
      P(WHE)=PT(WHE)-PT(MIX)*PW(MIX,IWIND)                              CAMS
      P(MIX)=PT(MIX)                                                    CAMS
```

```
      P(OTH)=100.-P(MIX)-P(WHE)                                      CAMS
 2610 CONTINUE                                                       CAMS
      PES=P(WHE)*XI(WHE)+P(MIX)*XI(MIX)+P(OTH)*XI(OTH)               CAMS
 2750 CONTINUE                                                       CAMS
C                                                                    CAMS
C   IF REPORT, GENERATE REPORT                                       CAMS
      IF(PRINTF.LE.0) GO TO 2700                                     CAMS
      DO 2800 I=1,4                                                  CAMS
      TITL(I)=HEAD(I,WINDOW)                                         CAMS
 2800 CONTINUE                                                       CAMS
      IDATE=IWIN(WINDOW,1)                                           CAMS
      PESTIM=PES                                                     CAMS
      TOT=PES-PT(WHE)                                                CAMS
      IF(IMODEL.EQ.2) P(OTH)=0.                                      CAMS
      IF(IUSE.GT.6)GO TO 2850                                        CAMS
      A1=(1.+TERTOT(WHE))*PW(WHE,WINDOW)                             CAMS
      A2=(1.+TERTOT(MIX))*PW(MIX,WINDOW)                             CAMS
      A3=(1.+TERTOT(OTH))*PW(OTH,WINDOW)                             CAMS
      A1=AMIN1(A1,1.)                                                CAMS
      A2=AMIN1(A2,1.)                                                CAMS
      A3=AMIN1(A3,1.)                                                CAMS
      A1=AMAX1(A1,0.)                                                CAMS
      A2=AMAX1(A2,0.)                                                CAMS
      A3=AMAX1(A3,0.)                                                CAMS
      ALOCAL= P(WHE)*A1 +P(MIX)*A2 +P(OTH)*A3                        CAMS
      TID=IPRIOR(IUSE)                                               CAMS
      ZERO1=0.                                                       CAMS
      ZERO2=0.                                                       CAMS
      IF(ALOCAL.EQ.0.)ZERO1=.000005                                  CAMS
      IF(PES.EQ.0.) ZERO2=.000005                                    CAMS
      TRAINA=((PES+ZERO2)/(ALOCAL+ZERO1))*100.                       CAMS
      TRAIND=(PES-ALOCAL)/(ALOCAL+ZERO1)*100.                        CAMS
      GO TO 2860                                                     CAMS
 2850 CONTINUE                                                       CAMS
      ALOCAL=PESTIM                                                  CAMS
      Z(1,1)=DELTA                                                   CAMS
      Z(2,1)=FLOAT(CROPD)                                            CAMS
      Z(1,2)=MULT(1)                                                 CAMS
      Z(2,2)=MULT(2)                                                 CAMS
      Z(3,2)=MULT(3)                                                 CAMS
      TID=99999999
      TRAINA=0.                                                      CAMS
      TRAIND=0.                                                      CAMS
 2860 CONTINUE                                                       CAMS
      IFIRST=IFIRST+1                                                CAMS
      CALL REPORT(2,IFIRST,IREP)                                     CAMS
C                                                                    CAMS
C  STORE VALUES IN OUTPUT RECORD                                     CAMS
 2700 CONTINUE                                                       CAMS
      IZULU(WINDOW)=IWIN(WINDOW,1)                                   CAMS
      PEST(WINDOW)=PES                                               CAMS
      PERR(WINDOW)=PES-PT(WHE)                                       CAMS
      GO TO 2100                                                     CAMS
 2101 CONTINUE                                                       CAMS
C                                                                    CAMS
C   WRITE TO OUTPUT FILE                                             CAMS
      COUN6=COUN4                                                    CAMS
      IREG6=IREG4                                                    CAMS
      IZONE6=IZONE4                                                  CAMS
      ISTRA6=ISTRA4                                                  CAMS
```

```
      ISUB6=ISUB4                                                       CAMS
      ISEG6=ISEG4                                                       CAMS
      PTRUE=PT(WHF)                                                     CAMS
      WRITE(CAMSF)COUN6,IREG6,IZONE6,ISTRA6,ISUB6,ISEG6,                CAMS
     1   PTRUE,(IZULU(I),PEST(I),PERR(I),I=1,4)                         CAMS
      NCAMSR=NCAMSR+1                                                   CAMS
      GO TO 2010                                                        CAMS
C                                                                       CAMS
C   DONE PROCESSING                                                     CAMS
 4000 CONTINUE                                                          CAMS
      IF(NFATAL.GT.0) RETURN                                            CAMS
      ITOT=LCAMSF-1                                                     CAMS
      WRITE(CAMSF)  ZZZZ,(IFILL,I=1,ITOT)                               CAMS
      REWIND CAMSF                                                      CAMS
      REWIND SEGTRU                                                     CAMS
      IF(IACQ.LE.0) REWIND ACQUIS                                       CAMS
      IF(IACQ.LE.0) REWIND CAMERR                                       CAMS
      IF(ICROPW.LE.0) REWIND CROPW                                      CAMS
      IF(ISIG.LE.0) REWIND SIGEXT                                       CAMS
      IF(IACQ .LE. 0)CALL TSAVE(0,-1,IBAD)
      RETURN                                                            CAMS
      END                                                               CAMS
```

```
      SUBROUTINE CAMSIN                                               CAMSIN
C        FILE DEFINITIONS AND RECORD LENGTHS                          FILFS
      COMMON  /FILES /                                                FILFS
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ        FILFS
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO       FILFS
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD       FILFS
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                    FILFS
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT FILFS
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF              FILFS
C                                                                     FILFS
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                        PAGECM
      COMMON  /PAGECM/                                                PAGECM
     1   MPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                       PAGECM
C                                                                     PAGECM
C        ARGUMENT LIST FOR ERROR PROCESSING                           ARGLST
      COMMON  /ARGLST/                                                ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                          ARGLST
      DIMENSION  IARG(10)                                             ARGLST
      EQUIVALENCE  ( IARG,ARG )                                       ARGLST
C                                                                     ARGLST
C        CAMS CONTROL CARD INPUT DATA                                 CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,     CAMSCM
     1   IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                   CAMSCM
      REAL MS                                                         CAMSCM
C                                                                     CAMSCM
      DIMENSION ICHK(4,2,15),CHKM(4,2,3),ISEQ(4,2)                    CAMSIN
      DIMENSION CHKG(4,2,2 ),CHKH(4,2,2),ISC(4,2)                     CAMSIN
      DATA ISC/10,10,10,11,12,12,12,13/                               CAMSIN
      DATA CHKM(1,1,1),CHKM(1,2,1),CHKM(2,1,1),CHKM(2,2,1),           CAMSIN
     1     CHKM(3,1,1),CHKM(3,2,1),CHKM(4,1,1),CHKM(4,2,1)
     2/1.,1.,1.,1.,1.,1.,1.,1./
      DATA ICHK(1,1,1),ICHK(1,2,1),ICHK(1,1,2),ICHK(1,2,2),           CAMSIN
     1   ICHK(1,1,3),ICHK(1,2,3),ICHK(1,1,4),ICHK(1,2,4),             CAMSIN
     1   ICHK(2,1,1),ICHK(2,2,1),ICHK(2,1,2),ICHK(2,2,2),             CAMSIN
     1   ICHK(2,1,3),ICHK(2,2,3),ICHK(2,1,4),ICHK(2,2,4),             CAMSIN
     1   ICHK(3,1,1),ICHK(3,2,1),ICHK(3,1,2),ICHK(3,2,2),             CAMSIN
     1   ICHK(3,1,3),ICHK(3,2,3),ICHK(3,1,4),ICHK(3,2,4),             CAMSIN
     1    ICHK(4,1,1),ICHK(4,2,1),ICHK(4,1,2),ICHK(4,2,2),            CAMSIN
     1   ICHK(4,1,3),ICHK(4,2,3),ICHK(4,1,4),ICHK(4,2,4)              CAMSIN
     1/1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,/
C                                                                     CAMSIN
C   READ IN AND CHECK CONTROL CARD                                    CAMSIN
      READ(INP,1000)IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,      CAMSIN
     1  ICAMS,ISEQ1                                                   CAMSIN
 1000 FORMAT(4I1,I2,I1,I1,66X,A4,I2)                                  CAMSIN
C                                                                     CAMSIN
C   ECHO OUT CONTROL CARD                                             CAMSIN
      CALL EJECT(8)                                                   CAMSIN
      WRITE(OUTP,1100)                                                CAMSIN
      WRITE(OUTP,1100)                                                CAMSIN
      WRITE(OUTP,1101)                                                CAMSIN
 1101 FORMAT(15X,                                                     CAMSIN
     1 50HC A M S   I N P U T   C O N T R O L   C A R D S             CAMSIN
      WRITE(OUTP,1100)                                                CAMSIN
      WRITE(OUTP,1100)                                                CAMSIN
 1100 FORMAT(2X)                                                      CAMSIN
      WRITE(OUTP,1200)                                                CAMSIN
 1200 FORMAT(3X,6HIMODEL,2X,6HIMULTI,2X,6HISIGEX,2X,5HISKIP,2X,5HITMAX, CAMSIN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1  2X,4HIREP,2X,5HIWIND)                                           CAMSIN
      WRITE(OUTP,1300) IMODEL,IMULTI,ISIGFX,ISKIP,ITMAX,IREP,IWIND,     CAMSIN
     1  ICAMS,ISEQ1                                                     CAMSIN
 1300 FORMAT(5X,I1,7X,I1,7X,I1,6X,I1,6X,I2,2X,2(4X,I1),5X,A4,I2)        CAMSIN
      WRITE(OUTP,1100)                                                  CAMSIN
C                                                                       CAMSIN
C   CHECK ID AND SEQ. NO.                                               CAMSIN
      IF(ICAMS.EQ.4HCAMS.AND.ISEQ1.EQ.1) GO TO 1                        CAMSIN
      NARG=2                                                            CAMSIN
      IARG(1)=ICAMS                                                     CAMSIN
      IARG(2)=ISEQ1                                                     CAMSIN
      CALL ERRMES(4HCAMS,6HCAMSIN,4,1)                                  CAMSIN
      GO TO 2                                                           CAMSIN
    1 CONTINUE                                                          CAMSIN
      IF(IWIND.EQ.0) IWIND=4                                            CAMSIN
      IARG(1)=ISEQ1                                                     CAMSIN
      IARG(2)=IMODEL                                                    CAMSIN
C                                                                       CAMSIN
C   CHECK CONTROL VALUES                                                CAMSIN
      IF(IMODEL.NE.1.AND.IMODEL.NE.2) CALL ERRMES(4HCAMS,6HCAMSIN,1,1)  CAMSIN
      IF(IMULTI.NE.0.AND.IMULTI.NE.1) IMULTI=1                          CAMSIN
      IF(ISIGFX.NE.0.AND.ISIGFX.NE.1) ISIGFX=1                          CAMSIN
      IF(ISKIP.NE.0.AND.ISKIP.NE.1) ISKIP=1                             CAMSIN
      IF(IREP.NE.0.AND.IREP.NE.1) IREP=1                                CAMSIN
      NARG=2                                                            CAMSIN
      IARG(2)=ITMAX                                                     CAMSIN
      IF(ITMAX.LT.0) CALL ERRMES(4HCAMS,6HCAMSIN,2,1)                   CAMSIN
      IARG(2)=IWIND                                                     CAMSIN
      IF(IWIND.LT.1.OR.IWIND.GT.4) CALL ERRMES(4HCAMS,6HCAMSIN,3,1)     CAMSIN
C*********************************************************************  CAMSIN
C                                                                       CAMSIN
C   READ IN AND CHECK MULTI-TEMPORAL SAMPLING MATRIX                    CAMSIN
    2 CONTINUE                                                          CAMSIN
      IFLAG=0                                                           CAMSIN
      INEXT=1                                                           CAMSIN
      CALL PAGER(9)                                                     CAMSIN
      WRITE(OUTP,2200) (I,I=5,15)                                       CAMSIN
 2200 FORMAT(3X,6HIGROUP,11(1H(,I2,1H),1X),3X,2HM2,5X,2HM3)             CAMSIN
      DO 20 I=1,2                                                       CAMSIN
      DO 21 J=1,4                                                       CAMSIN
      INEXT=INEXT+1                                                     CAMSIN
      READ(INP,2000) (ICHK(J,I,K),K=5,15),                              CAMSIN
     1               (CHKM(J,I,K),K=2,3),ICAMS,ISEQ(J,I)                CAMSIN
 2000 FORMAT(11I1,2X,2F4.3,53X,A4,I2)                                   CAMSIN
      IF(ICAMS.EQ.4HCAMS.AND.ISEQ(J,I).EQ.INEXT)  GO TO 22              CAMSIN
      IFLAG=1                                                           CAMSIN
      IARG(1)=ICAMS                                                     CAMSIN
      IARG(2)=ISEQ(J,I)                                                 CAMSIN
      CALL ERRMES(4HCAMS,6HCAMSIN,4,1)                                  CAMSIN
   22 CONTINUE                                                          CAMSIN
C                                                                       CAMSIN
C   ECHO OUT                                                            CAMSIN
      WRITE(OUTP,2100) (ICHK(J,I,K),K=5,15),                            CAMSIN
     1      (CHKM(J,I,K),K=2,3),ICAMS,ISEQ(J,I)                         CAMSIN
 2100 FORMAT(11X,10(I1,4X),I1,3X,2(F5.3,2X),A4,I2)                      CAMSIN
   21 CONTINUE                                                          CAMSIN
   20 CONTINUE                                                          CAMSIN
C                                                                       CAMSIN
C   MOVE INTO ARRAY                                                     CAMSIN
      NARG=3                                                            CAMSIN
```

```
      ITOT=3                                                            CAMSIN
      IWHER=0                                                           CAMSIN
      IF(IMODEL.EQ.2) ITOT=1                                            CAMSIN
      IF(IMODEL.EQ.2) IWHER=3                                           CAMSIN
      IF(IFLAG.EQ.1 .OR.IMULTI.EQ.1) GO TO 40                           CAMSIN
      DO 30 I=1,2                                                       CAMSIN
      DO 31 J=1,ITOT                                                    CAMSIN
      L=J+IWHER                                                         CAMSIN
      IGROUP(J,I,1)=1                                                   CAMSIN
      IGROUP(J,I,2)=1                                                   CAMSIN
      IGROUP(J,I,3)=1                                                   CAMSIN
      IGROUP(J,I,4)=1                                                   CAMSIN
      DO 32 K=5,15                                                      CAMSIN
      IARG(1)=ISEQ(L,I)                                                 CAMSIN
      IARG(2)=K                                                         CAMSIN
      IARG(3)=ICHK(L,I,K)                                               CAMSIN
      IF(ICHK(L,I,K).LT.1.OR.ICHK(L,I,K).GT.3) CALL  ERRMES(4HCAMS,     CAMSIN
     1   6HCAMSIN,7,1)                                                  CAMSIN
      IGROUP(J,I,K)=ICHK(L,I,K)                                         CAMSIN
   32 CONTINUE                                                          CAMSIN
      MS(J,I,1)=1.                                                      CAMSIN
      DO 33 K=2,3                                                       CAMSIN
      IARG(2)=K                                                         CAMSIN
      ARG(3)=CHKM(L,I,K)                                                CAMSIN
      IF(CHKM(L,I,K).LE.0.0  .OR. CHKM(L,I,K).GE.CHKM(L,I,K-1))         CAMSIN
     1     CALL ERRMES(4HCAMS,6HCAMSIN,6,1)                             CAMSIN
      MS(J,I,K)=CHKM(L,I,K)                                             CAMSIN
   33 CONTINUE                                                          CAMSIN
   31 CONTINUE                                                          CAMSIN
   30 CONTINUE                                                          CAMSIN
C*********************************************************************** CAMSIN
C                                                                       CAMSIN
C   READ IN AND CHECK CROP CALENDAR                                     CAMSIN
   40 CONTINUE                                                          CAMSIN
      CALL PAGER(6)                                                     CAMSIN
      WRITE(OUTP,1100)                                                  CAMSIN
      WRITE(OUTP,4000)                                                  CAMSIN
 4000 FORMAT(4X,3(2HG1,6X,2HG2,6X,2HH1,6X,2HH2,6X))                      CAMSIN
      DO 41 I=1,2                                                       CAMSIN
      INEXT=INEXT+1                                                     CAMSIN
      READ(INP,4040)(CHKG(1,I,J),J=1,2),(CHKH(1,I,J),J=1,2),            CAMSIN
     1                 (CHKG(2,I,J),J=1,2),(CHKH(2,I,J),J=1,2),         CAMSIN
     1                 (CHKG(3,I,J),J=1,2),(CHKH(3,I,J),J=1,2)          CAMSIN
     1                 ,ICAMS,ISEQ(I,1)                                 CAMSIN
 4040 FORMAT(2(F6.3,F6.2,F6.3,F6.2,1X),2(F6.3,F6.2),A4,I2)               CAMSIN
      IF(ICAMS.EQ.4HCAMS.AND.ISEQ(I,1).EQ.INEXT) GO TO 42               CAMSIN
      IFLAG=2                                                           CAMSIN
      NARG=2                                                            CAMSIN
      IARG(1)=ICAMS                                                     CAMSIN
      IARG(2)=ISEQ(I,1)                                                 CAMSIN
      CALL ERRMES(  4HCAMS,6HCAMSIN,4,1)                                CAMSIN
   42 CONTINUE                                                          CAMSIN
      WRITE(OUTP,4100) (CHKG(1,I,J),J=1,2),(CHKH(1,I,J),J=1,2),         CAMSIN
     1                 (CHKG(2,I,J),J=1,2),(CHKH(2,I,J),J=1,2),         CAMSIN
     1                 (CHKG(3,I,J),J=1,2),(CHKH(3,I,J),J=1,2)          CAMSIN
     1                 ,ICAMS,ISEQ(I,1)                                 CAMSIN
 4100 FORMAT(2X,3(F6.3,2X,F6.2,2X,F6.3,2X,F6.2,2X),2X,A4,I2)             CAMSIN
      INEXT=INEXT+1                                                     CAMSIN
      READ(INP,4200) (CHKG(4,I,J),J=1,2),(CHKH(4,I,J),J=1,2),ICAMS,ISEQ(ICAMSIN
 4200 FORMAT(2(F6.3,F6.2),50X,A4,I2)                                    CAMSIN
```

```
      IF(ICAMS.EQ.4HCAMS.AND.ISEQ1.EQ.INEXT) GO TO 43                   CAMSIN
      IFLAG=2                                                           CAMSIN
      IARG(1)=ICAMS                                                     CAMSIN
      IARG(2)=ISEQ1                                                     CAMSIN
      CALL   ERRMES (4HCAMS,6HCAMSIN,4,1)                               CAMSIN
   43 CONTINUE                                                          CAMSIN
      WRITE(OUTP,4300)                                                  CAMSIN
     1                (CHKG(4,I,J),J=1,2),(CHKH(4,I,J),J=1,2),ICAMS,ISEQ1 CAMSIN
 4300 FORMAT(2X,2(F6.3,2X,F6.2,2X),66X,A4,I2)                           CAMSIN
   41 CONTINUE                                                          CAMSIN
C                                                                       CAMSIN
C     MOVE INTO ARRAY                                                   CAMSIN
      IF(IFLAG.EQ.2.) GO TO 50                                          CAMSIN
      NARG=2                                                            CAMSIN
      DO 51 I=1,2                                                       CAMSIN
      DO 52 J=1,ITOT                                                    CAMSIN
      L=J+IWHER                                                         CAMSIN
      IARG(1)=ISEQ(I,1) + IWHER/4                                       CAMSIN
      ARG(2)=CHKG(L,I,1)                                                CAMSIN
      IF(ABS(CHKG(L,I,1)).GE.10.0 ) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKH(L,I,1)                                                CAMSIN
      IF(ABS(CHKH(L,I,1)).GE.10.0 ) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKG(L,I,2)                                                CAMSIN
      IF(ABS(CHKG(L,I,2)).GE.100.0) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKH(L,I,2)                                                CAMSIN
      IF(ABS(CHKH(L,I,2)).GE.100.0) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      G(J,I,1)=CHKG(L,I,1)                                              CAMSIN
      G(J,I,2)=CHKG(L,I,2)                                              CAMSIN
      H(J,I,1)=CHKH(L,I,1)                                              CAMSIN
      H(J,I,2)=CHKH(L,I,2)                                              CAMSIN
   52 CONTINUE                                                          CAMSIN
   51 CONTINUE                                                          CAMSIN
   50 CONTINUE                                                          CAMSIN
      RETURN                                                            CAMSIN
      END                                                               CAMSIN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(ICAMS.EQ.4HCAMS.AND.ISEQ1.EQ.INEXT) GO TO 43                   CAMSIN
      IFLAG=2                                                           CAMSIN
      IARG(1)=ICAMS                                                     CAMSIN
      IARG(2)=ISEQ1                                                     CAMSIN
      CALL   ERRMES (4HCAMS,6HCAMSIN,4,1)                               CAMSIN
   43 CONTINUE                                                          CAMSIN
      WRITE(OUTP,4300)                                                  CAMSIN
     1          (CHKG(4,I,J),J=1,2),(CHKH(4,I,J),J=1,2),ICAMS,ISEQ1     CAMSIN
 4300 FORMAT(2X,2(F6.3,2X,F6.2,2X),66X,A4,I2)                           CAMSIN
   41 CONTINUE                                                          CAMSIN
C                                                                       CAMSIN
C  MOVE INTO ARRAY                                                      CAMSIN
      IF(IFLAG.EQ.2 ) GO TO 50                                          CAMSIN
      NARG=2                                                            CAMSIN
      DO 51 I=1,2                                                       CAMSIN
      DO 52 J=1,ITOT                                                    CAMSIN
      L=J+INHER                                                         CAMSIN
      IARG(1)=ISEQ(I,1) + INHER/4                                       CAMSIN
      ARG(2)=CHKG(L,I,1)                                                CAMSIN
      IF(ABS(CHKG(L,I,1)).GE.10.0 ) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKH(L,I,1)                                                CAMSIN
      IF(ABS(CHKH(L,I,1)).GE.10.0 ) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKG(L,I,2)                                                CAMSIN
      IF(ABS(CHKG(L,I,2)).GE.100.0) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      ARG(2)=CHKH(L,I,2)                                                CAMSIN
      IF(ABS(CHKH(L,I,2)).GE.100.0) CALL ERRMES(4HCAMS,6HCAMSIN,5,1)    CAMSIN
      G(J,I,1)=CHKG(L,I,1)                                              CAMSIN
      G(J,I,2)=CHKG(L,I,2)                                              CAMSIN
      H(J,I,1)=CHKH(L,I,1)                                              CAMSIN
      H(J,I,2)=CHKH(L,I,2)                                              CAMSIN
   52 CONTINUE                                                          CAMSIN
   51 CONTINUE                                                          CAMSIN
   50 CONTINUE                                                          CAMSIN
      RETURN                                                            CAMSIN
      END                                                               CAMSIN
```

```
FOR,IS CAS
      SUBROUTINE CAS                                                    CAS
C        MAIN DRIVER FOR CAS SIMULATOR                                  CAS
C                                                                       CAS
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                      CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)   ,PPRUTS(5,2)   ,YPRUTS(3,2)   CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)    CASCM
     3  ,WPRIOR(4)     ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)         CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW    ,IBW    ,WINDOW,IPD    ,IPP    ,PPDATE,NREGS  CASFLG
     2  ,NZTOT  ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS       CASFLG
     3  ,ENDC   ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
     4  ,LDS1   ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR        CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CAS DATA SETS 1,2, AND 3                                       DSET1
      COMMON  /DSET1 /                                                  DSET1
     1   ISUBST,TWAK  ,HWAK  ,EWAK  ,M1K   ,CT1K  ,ANALVK,EPWK  ,EPW2K  JULY76
     2  ,SMPKPI,SUMPK2,SUMPK ,KSUB  ,NCLASS                             JULY76
      REAL  M1K , M2K                                                   JULY76
      DIMENSION  DSET1(14), DSET2(14), DSET3(6)                         JULY76
      EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                         DSET1
     1  , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                               DSET1
C                                                                       DSET1
```

```
C          CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                    DSET4
       COMMON  /DSET4 /                                                   DSET4
      1    STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                JULY76
      2   ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                    JULY76
      3 ,TWAS3,HWAS3,XCT3S
      4   ,XYS    ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS   ,ANVARS             JULY76
      5 ,FILL4(57)
       INTEGER  STRATA                                                    JULY76
       DIMENSION  DSET4(24), DSET5(7), DSET6(3)                           JULY76
       EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )    DSET4
C                                                                         DSET4
C          CAS DATA SET 7 (AT ZONE LEVEL)                                 DSET7
       COMMON  /DSET7 /                                                   DSET7
      1    ZONE   ,HWAZ2 ,EZ     ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3 JULY76
      2   ,ESTVZ  ,HWAZ12                                                 JULY76
      3   ,M1K2CL(10)    ,EPWCL(10)     ,EPW2CL(10)    ,PKPICL(10)        JULY76
      4   ,PK2CL(10)     ,PKCL(10)      ,SSQ(10)                          JULY76
       INTEGER  ZONE                                                      JULY76
       REAL  M1K2KZ, M1K2CL                                               JULY76
       DIMENSION  DSET7(81)                                               JULY76
       EQUIVALENCE  ( DSET7,ZONE )                                        DSET7
C                                                                         DSET7
C          CAS DATA SET 8 (AT REGION LEVEL)                               DSET8
       COMMON  /DSET8 /                                                   DSET8
      1    REGION,HWAR2 ,ER     ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR JULY76
      2   ,M1M2ZR,FILL8(71)                                               JULY76
       INTEGER  REGION                                                    JULY76
       REAL  M1K2KR                                                       JULY76
       DIMENSION  DSET8(10)                                               JULY76
       EQUIVALENCE  ( DSET8,REGION )                                      DSET8
C                                                                         DSET8
C          CAS DATA SET 9 (AT COUNTRY LEVEL)                              DSET9
       COMMON  /DSET9 /                                                   DSET9
      1    COUNTR,HWAC2 ,EC     ,M1K2KC,ANALVC,M1M2ZC,HWAC1 ,EWAC1 ,ESTVC JULY76
       INTEGER  COUNTR                                                    JULY76
       REAL  M1K2KC ,                                                     JULY76
       DIMENSION  DSET9(9)                                                JULY76
       EQUIVALENCE  ( DSET9,COUNTR )                                      DSET9
C                                                                         DSET9
C          CAS DATA SET 10 (STRATA DATA -- FINAL PASS)                    JULY76
       COMMON  /DSET10/                                                   JULY76
```

```
     1   HWAS   ,TWAS   ,EWAS   ,AERRS ,AVARS ,TPRODS,EPRODS,PRERRS,PRVARS JULY76
     2  ,YS     ,ESTYS ,YERRS ,M1JS   ,M2JS   ,CT1S   ,CT2S   ,CT3S   ,ANAVS  JULY76
     3  ,ANPRVS,ES                                                          JULY76
      REAL  M1JS , M2JS                                                   JULY76
      DIMENSION  DSET10(20)                                               JULY76
      EQUIVALENCE  ( DSET10,HWAS )                                        JULY76
C                                                                         DSET10
C        CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                        JULY76
      COMMON  /DSET11/                                                    DSET11
     1   HWAZ   ,TWAZ   ,EWAZ   ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ    ,EYZ    ,YERRZ ,M1Z    ,M2Z    ,CT1Z   ,CT2Z   ,CT3Z   ,ANAVZ  DSET11
     3  ,ANPRVZ                                                           DSET11
      REAL  M1Z , M2Z                                                     DSET11
      DIMENSION  DSET11(19)                                               DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                        DSET11
C                                                                         DSET11
C        CAS DATA SET 12 (REGION DATA -- FINAL PASS)                      JULY76
      COMMON  /DSET12/                                                    DSET12
     1   HWAR   ,TWAR   ,EWAR   ,AERRR ,AVARR ,TPRODR,EPRODR,PRERRR,PRVARR DSET12
     2  ,TYR    ,EYR    ,YERRR ,M1R    ,M2R    ,CT1R   ,CT2R   ,CT3R   ,ANAVR  DSET12
     3  ,ANPRVR                                                           DSET12
      REAL  M1R , M2R                                                     DSET12
      DIMENSION  DSET12(19)                                               DSET12
      EQUIVALENCE  ( DSET12,HWAR )                                        DSET12
C                                                                         DSET12
C        CAS DATA SET 13 (COUNTRY DATA -- FINAL PASS)                     JULY76
      COMMON  /DSET13/                                                    DSET13
     1   HWAC   ,TWAC   ,EWAC   ,AERRC ,AVARC ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2  ,TYC    ,EYC    ,YERRC ,M1C    ,M2C    ,CT1C   ,CT2C   ,CT3C   ,ANAVC  DSET13
     3  ,ANPRVC,CLEWA ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC                 DSET13
      REAL  M1C , M2C                                                     DSET13
      DIMENSION  DSET13(25)                                               DSET13
      EQUIVALENCE  ( DSET13,HWAC )                                        DSET13
C                                                                         DSET13
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)         IXCDSF
      COMMON  /IXCDSF/                                                    IXCDSF
     1   IXCDSF(1),LIXCDS
C                                                                         IXCDSF
C        LEM CONTROL CARD INPUT DATA                                      LEMCM
      COMMON  /LEMCM /                                                    LEMCM
     1   TITLE(10)    ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
```

```
    2  ,ENDR   ,ENDZ   ,ISTG   ,ICAMS  ,IYES   ,IACQ   ,ICLASS,ISEXT  ,ISCC    LEMCM
    3  ,ICAS2  ,ICAS3  ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
    4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
    5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                       LEMCM
     DIMENSION  RSEED(7)                                                LEMCM
     DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5        LEMCM
    1  ,RSEED6,RSEED7                                                   LEMCM
     EQUIVALENCE  ( RSEED,RSEED1 )                                      LEMCM
     INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                          LEMCM
C                                                                       LEMCM
C        SEGMENT DATA FROM CAMS OUTPUT FILE (CAMSF)                     SEGDTA
     COMMON  /SEGDTA/                                                   SEGDTA
    1    IDSEGT(5)    ,ISEG   ,TPWKI  ,ZACDAY(4)    ,EPWKI(4)           SEGDTA
    2  ,ERRPWI(4)    ,ESTPWI                                            SEGDTA
     INTEGER  ZACDAY                                                    SEGDTA
C                                                                       SEGDTA
C        SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                     SSHDTA
     COMMON  /SSHDTA/                                                   SSHDTA
    1    COUN2  ,IREG2  ,IZONE2,ISTRA2,ISUBS2,NSEG   ,IDSEG  ,GRPNO  ,HISTPW SSHDTA
    2  ,AREAK  ,PWK    ,NAGR   ,NA     ,DELTPW,DELTPM,CV1    ,CV2    ,CV3    SSHDTA
    3  ,CV4    ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
     INTEGER  GRPNO , CLASS , RDSSH                                     JULY76
     DIMENSION  SSHDTA(39)                                              JULY76
     EQUIVALENCE  ( SSHDTA, COUN2 )                                     SSHDTA
C                                                                       SSHDTA
C        STATISTICAL INFORMATION FOR LEM                                STATS
     COMMON  /STATS /                                                   STATS
    1    ITER   ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR             STATS
     EQUIVALENCE  ( NT,ITER )                                           STATS
C                                                                       STATS
C        SUMMARY DATA FOR REPORTS                                       SUMDTA
     COMMON  /SUMDTA/                                                   SUMDTA
    1    CVAEPT,CVEPTA,SDPER ,CVPEPT,CVEPTP,CSUMR(18,18)                SUMDTA
C                                                                       SUMDTA
C        YIELD DATA FROM YESOUT FILE                                    YESDTA
     COMMON  /YESDTA/                                                   YESDTA
    1    YSTR   ,IZPROD(6)    ,YSCI(6)      ,VSYCI(6)                   YESDTA
    2  ,RDYES  ,NYESPP                                                  YESDTA
     INTEGER  RDYES                                                     YESDTA
C                                                                       YESDTA
C                                                                       CAS
```

```
C          PERFORM PROGRAM INITIALIZATION TASKS                                   CAS
       CALL CASINT                                                                CAS
       IF ( NFATAL .NE. 0 )  GO TO 900                                            CAS
C                                                                                 CAS
C          BIOWINDOW LOOP                                                         CAS
       IBW= 0                                                                     CAS
  210  IBW= IBW + 1                                                               CAS
       IF ( BWIND(IBW) .EQ. 0 )  GO TO 250                                        CAS
C          PROCESS NEXT BIOWINDOW                                                 CAS
       CALL CASPP                                                                 CAS
       IF ( NFATAL .NE. 0 )  GO TO 900                                            CAS
  250  IF ( IBW .LT. 4 ) GO TO 210                                                CAS
C                                                                                 CAS
C          PREDICTION DATE LOOP                                                   CAS
  300  PPFLG= 1                                                                   CAS
       IPD= 0                                                                     CAS
  310  IPD= IPD + 1                                                               CAS
       PPDATE= PRDATE(IPD)                                                        CAS
       IF ( PPDATE .EQ. 0 )  GO TO 400                                            CAS
C          PROCESS NEXT PREDICTION POINT                                          CAS
       CALL CASPP                                                                 CAS
       IF ( NFATAL .NE. 0 )  GO TO 900                                            CAS
       IF ( IPD .LT. NPDATE )  GO TO 310                                          CAS
C                                                                                 CAS
C          WHEN PRINTF .NE. 0, PRINT COUNTRY SUMMARY REPORT                       CAS
  400  IF ( PRINTF .NE. 0 )  CALL SUMREP                                          CAS
  900  RETURN                                                                     CAS
       END                                                                        CAS
```

@ FLT CASBLD,1,760427, 59062   , 1

```
      BLOCK DATA CASBLD                                                 CASBLD
C                                                                       CASBLD
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)       IXCDSF
      COMMON  /IXCDSF/                                                  IXCDSF
     1   IXCDSF(  1)  ,LIXCDS                                           IXCDSF
C                                                                       IXCDSF
C                                                                       CASBLD
      DATA  LIXCDS  / 388 /                                             CASBLD
C                                                                       CASBLD
      END                                                               CASBLD
```

```
      SUBROUTINE CASER1(INO)                                            CASER1
C                                                                       CASER1
C     THIS SUBROUTINE CONTAINS THE INPUT ERROR MESSAGES FOR CAS         CASER1
C                                                                       CASER1
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUIS(4,2)  ,PPRUIS(5,2)  ,YPRUIS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)    CASCM
     3  ,NPRIOR(4)     ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)         CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,NPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SURHST,LSURH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO        FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SURHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C                                                                       CASER1
C                                                                       CASER1
      GO TO (10,20,30,40,50,60,70),INO                                  CASER1
   10 CONTINUE                                                          CASER1
      WRITE(OUTP,1000) IARG(1),IARG(2)                                  CASER1
      RETURN                                                            CASER1
   20 CONTINUE                                                          CASER1
      WRITE(OUTP,2000)  NHISTY                                          CASER1
      RETURN                                                            CASER1
   30 CONTINUE                                                          CASER1
      WRITE(OUTP,3000)  HH                                              CASER1
      HH= 99999                                                         CASER1
      RETURN                                                            CASER1
   40 CONTINUE                                                          CASER1
      WRITE(OUTP,4000)  NPRIOR                                          CASER1
      RETURN                                                            CASER1
   50 CONTINUE                                                          CASER1
      WRITE(OUTP,5000)                                                  CASER1
      RETURN                                                            CASER1
   60 CONTINUE                                                          CASER1
      WRITE(OUTP,6000)IARG(1),IARG(2),IARG(3),IARG(1),IARG(2),IARG(3)   CASER1
      RETURN                                                            CASER1
   70 CONTINUE                                                          CASER1
      WRITE(OUTP,7000)                                                  CASER1
      RETURN                                                            CASER1
 1000 FORMAT(2X,                                                        CASER1
     1 57HIMPROPER LABEL AND SEQUENCE NUMBER ON A CAS CONTROL CARD.     CASER1
     1   //2X,21HLABEL AND SEQ. NO. = ,A4,2X,I2)                        CASER1
 2000 FORMAT(2X,9HNHISTY = ,I3,17H IS OUT OF RANGE.    ,                CASER1
     1   //2X,25H(1 .LE. NHISTY .LE. 20)    )                           CASER1
 3000 FORMAT(2X,5HHH = ,I3,16H IS OUT OF RANGE                          CASER1
     1   //2X,19H(3 .LE. HH .LE. 99)   )                                CASER1
```

```
      4000 FORMAT(2X,37HILLEGAL WINDOW SPECIFIED IN WPRIOR = ,            CASFR1
          1  3(I1,2H, ),I1  /2X,30HEACH WINDOW MUST BE 1-4 OR 0)    )     CASFR1
      5000 FORMAT(2X,30HALL ENTRIES IN WPRIOR ARE ZERO   )                CASFR1
      6000 FORMAT(2X,24HILLEGAL PREDICTION DATE ,2(I2,1H/),I2,11H SPECIFIED. CASFR1
          1  /2X,7HYEAR = ,I2,26HMUST BE .GE. 64, MONTH =                 CASFR1
          1  I2,21H MUST BE 1-12, DAY = ,I2,14H MUST BE 1-31.  )          CASFR1
      7000 FORMAT(2X,                                                     CASFR1
          1 53HPREDICTION DATES NOT IN ASCENDING ORDER OR DUPLICATES   )  CASFR1
           END                                                            CASFR1
```

```
FOR,IS CASER2
      SUBROUTINE CASER2 (ICODE)                                         CASER2
C        PRINTS PROCESSING ERROR MESSAGES FOR CAS SIMULATOR             CASER2
C                                                                       CASER2
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
      COMMON  /DSET4 /  STRATA                                          CASER2
      COMMON  /DSET7 /  ZONE                                            CASER2
      COMMON  /DSET8 /  REGION                                          CASER2
      INTEGER  STRATA, ZONE, REGION                                     CASER2
C                                                                       CASER2
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
```

```
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                     SSHDTA
      COMMON  /SSHDTA/                                                  SSHDTA
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                    JULY76
      DIMENSION  SSHDTA(39)                                             JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                    SSHDTA
C                                                                       SSHDTA
C        SEGMENT DATA FROM CAMS OUTPUT FILE (CAMSF)                     SEGDTA
      COMMON  /SEGDTA/                                                  SEGDTA
     1   IDSEGT(5)     ,ISEG  ,TPWKI ,ZACDAY(4)     ,EPWKI(4)           SEGDTA
     2  ,ERRPWI(4)     ,ESTPWI                                          SEGDTA
      INTEGER  ZACDAY                                                   SEGDTA
C                                                                       SEGDTA
C                                                                       CASER2
      DATA  MXMES  / 19 /                                               JULY76
C                                                                       CASER2
C                                                                       CASER2
      IMES= ICODE                                                       CASER2
      IF ( IMES .LT. 1 .OR. IMES .GT. MXMES )  GO TO 800                CASER2
C                                                                       CASER2
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170 CASER2
     1  ,180,190)  , IMES                                               JULY76
C                                                                       CASER2
 10   WRITE (OUTP,1)  IARG(1),ARG(2)                                    CASER2
 1    FORMAT (38HODIVISION BY ZERO NOT ALLOWED.  EQN. (,I3,4H),  ,A6,   CASER2
     1  5H = 0.)                                                        CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 20   WRITE (OUTP,2)                                                    CASER2
 2    FORMAT (105HOIF NT = 1, VARIANCE ERRORS AND CONFIDENCE LEVELS CANNCASER2
     1OT BE COMPUTED AND WILL BE ARBITRARILY SET TO ZERO.)              CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 30   IARG(2)= STARTR                                                   CASER2
```

```
      GO TO 45                                                        CASER2
 40   IARG(2)= STARTZ                                                 CASER2
 45   WRITE (OUTP,4)  ARG(1),IARG(2),ARG(3)                           CASER2
 4    FORMAT (10HOSTARTING ,A6,I5,15H  NOT FOUND ON A6,6H FILE.)      CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
 50   IARG(2)= ENDR                                                   CASER2
      GO TO 65                                                        CASER2
 60   IARG(2)= ENDZ                                                   CASER2
 65   WRITE (OUTP,6)  ARG(1),IARG(2),ARG(3)                           CASER2
 6    FORMAT (8HOENDING ,A6,I5,15H  NOT FOUND ON ,A6,6H FILE. )       CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
C                                                                     CASER2
 70   WRITE (OUTP,7)  REGION, ZONE, STRATA, NRYES                     CASER2
 7    FORMAT (49HOZERO PREDICTION DATES ON YESOUT FILE FOR REGION ,I4 CASER2
     1  ,7H, ZONE ,I4,9H, STRATA ,I4,15H  (DATA RECORD ,I4,1H))       CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
 80   WRITE (OUTP,8)  GRPNO,IREG2,IZONE2,ISTRA2,ISUBS2,NRSSH          CASER2
 8    FORMAT (22HOILLEGAL GROUP NUMBER ,I3,30H  FROM SUBHST FILE FOR REGCASER2
     1ION ,I4,7H, ZONE ,I4,10H, STRATUM ,I4,13H, SUBSTRATUM ,I4       CASER2
     2  /15H  (DATA RECORD ,I4,1H))                                   CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
 90   WRITE (OUTP,9)  NAGR,NA,GRPNO                                   CASER2
 9    FORMAT  (8HONAGR = ,I4,10H  OR NA = ,I4,42H  FROM FILE SUBHST ARE CASER2
     1ZERO. GROUP NUMBER ,I3,17H IS CHANGED TO 3. )                   CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
 100  WRITE (OUTP,910)NRYES,REGION,ZONE,STRATA,NRSSH,IREG2,IZONE2,ISTRA2CASER2
 910  FORMAT (47HOINCONSISTENCY BETWEEN YESOUT AND SUBHST FILES.//    CASER2
     1  8X,30HRECORD  REGION   ZONE   STRATA/8H YESOUT ,I5,3I8/       CASER2
     2  8H SUBHST ,I5,3I8)                                            CASER2
      GO TO 900                                                       CASER2
C                                                                     CASER2
 110  WRITE (OUTP,11)  NRSSH,IREG2,IZONE2,ISTRA2,ISUBS2,NRCAMS,IDSEGT(2)CASER2
     1  ,IDSEGT(3),IDSEGT(4),IDSEGT(5)                                CASER2
 11   FORMAT (46HOINCONSISTENCY BETWEEN SUBHST AND CAMSF FILES.//     CASER2
     1  8X,41HRECORD  REGION   ZONE   STRATA  SUBSTRATA               CASER2
     2  /8H SUBHST ,I5,3I8,I10/8H CAMSF  ,I5,3I8,I10)                 CASER2
```

```
      GO TO 900                                                         CASER2
C                                                                       CASER2
 120  WRITE (OUTP,12)  IARG(1)                                          CASER2
 12   FORMAT (61HOERROR RETURN FROM BETA DISTRIBUTION SUBROUTINE. ERROR CASER2
     1FLAG = I3)                                                        CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 130  WRITE (OUTP,13)  ISUBS2,ISTRA2,IZONE2,IREG2,NRSSH                 CASER2
 13   FORMAT (26HONO SEGMENTS IN SUBSTRATA ,I4,9H, STRATA ,I4,7H, ZONE  CASER2
     1  I4,9H, REGION I4,17H  (SUBHST RECORD ,I4,1H))                   CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 140  WRITE (OUTP,14)                                                   CASER2
 14   FORMAT (52HOZERO OR NEGATIVE DIVISOR IN COMPUTING TAU2S, SIGM2S/  CASER2
     1  14H(EQS. 93D-93F) )                                             CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 150  WRITE (OUTP,15)  HISTPW,ISUBS2,ISTRA2,IZONE2,IREG2                CASER2
 15   FORMAT (22HOWARNING...  HIST PW =F6.2,16H  FOR SUBSTRATA ,I4      CASER2
     1  ,9H, STRATA ,I4,7H, ZONE ,I4,9H, REGION ,I4/                    CASER2
     2  24H GROUP NO. CHANGED TO 3.)                                    CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 160  WRITE (OUTP,16)  IPD,PPDATE,STRATA,ZONE,REGION,NRYES              CASER2
 16   FORMAT (24HOINPUT PREDICTION DATE (,I2,4H) = I4/                  CASER2
     1  54H  .LT. ALL PREDICTION DATES ON YESOUT FILE FOR STRATA ,I4,   CASER2
     2  7H, ZONE ,I4,9H, REGION ,I4,10H  (RECORD ,I4,1H))               CASER2
      GO TO 900                                                         CASER2
 170  WRITE (OUTP,17)                                                   CASER2
 17   FORMAT  (58HOTOO MANY MONTE CARLO ITERATIONS FOR CAS DISTRIBUTION CASER2
     1FILE )                                                            CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 180  WRITE (OUTP,18)  ARG(1),IARG(2),ARG(3),ARG(4)                     CASER2
 18   FORMAT (2HO ,A6,9H  IN EQ. ,I4,3H = ,E13.6,15H  REF. VALUE = ,    CASER2
     1  E13.6)                                                          CASER2
      GO TO 900                                                         CASER2
C                                                                       CASER2
 190  WRITE (OUTP,19)  IARG(1),ZONE,REGION,IARG(2)                      JULY76
 19   FORMAT (40HOLESS THAN 2 ACQUIRED SEGMENTS IN CLASS I3,9H  OF ZONE JULY76
     1  I4,8H  REGION,I4,24H.  NO. OF ACQ. SEGMENTS=I3)                 JULY76
```

```
      GO TO 900                                                              JULY76
C                                                                            JULY76
C                                                                            CASER2
  800 WRITE (OUTP,801)  IMES                                                 CASER2
  801 FORMAT  (59H0 SUBROUTINE CASER2 CALLED WITH ILLEGAL ERROR MESSAGE      CASER2
     1CODE ,I4)                                                              CASER2
  900 RETURN                                                                 CASER2
      END                                                                    CASER2
```

```
      SUBROUTINE CASIN                                                  CASIN
C                                                                       CASIN
C  THIS SUBROUTINE READS IN AND CHECKS THE CAS CONTROL CARDS            CASIN
C                                                                       CASIN
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF   ,PRDCF ,APRUIS(4,2)  ,PPRUIS(5,2)  ,YPRUIS(3,2)   CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT  ,AUNITS,DISTFF,BWIND(4)     CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C         DATA BLOCK FOR CAS CUMULATIVE FILE                            CASCUM
C         CAS DATA SETS 14, 15, 16, AND 17                              CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C         DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)            CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFR )                                     CASDSB
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1  , ( PIK,CASDSB(121) )                                           CASDSB
C                                                                       CASDSB
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C         DATA BLOCK FOR CAS DISTRIBUTION FILE                          IXDISF
      COMMON  /IXDISF/                                                  IXDISF
     1   IXDISF(  1), LIXDIS                                            IXDISF
C        NOTE... 506 ONLY ALLOWS UP TO 8 PREDICTION POINTS INCLUDING    IXDISF
C        WINDOWS ( 506 = 1 + 1 + 8*63, INDEX + HEADER + 8 PRED. PTS.)IXDISF
C                                                                       IXDISF
C         LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASF ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISIG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYH,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
```

```
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASOR              STATS
      EQUIVALENCE  ( NI,ITER )                                          STATS
C                                                                       STATS
C                                                                       CASIN
C  READ IN FIRST CONTROL CARD                                           CASIN
      READ(INP,1000) NHISTY,HH,TOPT,AUNITS,DISTFF,BWIND,WPRIOR,APREP,   CASIN
     1   IARG(1),IARG(2)                                                CASIN
 1000 FORMAT(14I3,32X,A4,I2)                                            CASIN
      NSTTL=0                                                           CASIN
C                                                                       CASIN
C   ECHO OUT FIRST CONTROL CARD                                         CASIN
      CALL EJECT(12)                                                    CASIN
      WRITE(OUTP,2000)                                                  CASIN
 2000 FORMAT(2X)                                                        CASIN
      WRITE(OUTP,2000)                                                  CASIN
      WRITE(OUTP,3000)                                                  CASIN
 3000 FORMAT(15X,48HC A S   I N P U T   C O N T R O L   C A R D S      )CASIN
      WRITE(OUTP,2000)                                                  CASIN
      WRITE(OUTP,2000)                                                  CASIN
      WRITE(OUTP,4000) (I,I=1,4),(I,I=1,4)                              CASIN
 4000 FORMAT(3X,6HNHISTY,3X,2HHH,2X,4HTOPT,2X,6HAUNITS,2X,6HDISTFF,2X,  CASIN
     1  5HBWIND,4(1H(,I1,3H)  ),6HWPRIOR,4(1H(,I1,3H)  ),5HAPREP)       CASIN
      WRITE(OUTP,5000)NHISTY,HH,TOPT,AUNITS,DISTFF,BWIND,WPRIOR,APREP,  CASIN
     1  IARG(1),IARG(2)                                                 CASIN
 5000 FORMAT(4X,I3,4X,I3,2X,I3,4X,I3,5X,I3,8X,4(I3,2X),6X,4(I3,2X),1X,  CASIN
     1  I3,3X,A4,I2)                                                    CASIN
      IF(IARG(1).EQ.3HCAS.AND.IARG(2).EQ.1) GO TO 10                    CASIN
      CALL ERRMES(3HCAS,5HCASIN,1,1)                                    CASIN
      GO TO 200                                                         CASIN
C                                                                       CASIN
C   CHECK FOR ERRORS                                                    CASIN
   10 CONTINUE                                                          CASIN
      IF(NHISTY.LT.1.OR.NHISTY.GT.20) CALL ERRMES(3HCAS,5HCASIN,2,1)    CASIN
      IF(HH.LT.3.OR.HH.GT.99) CALL ERRMES(3HCAS,5HCASIN,3,0)            CASIN
      IF(TOPT.NE.0) TOPT=1                                              CASIN
      IF(AUNITS.NE.0) AUNITS=1                                          CASIN
      IF(DISTFF.NE.0) DISTFF=1                                          CASIN
      IF(APREP.NE.0) APREP=1                                            CASIN
      NBW=0                                                             CASIN
      DO 20 I=1,4                                                       CASIN
      IF(BWIND(I).NE.0) BWIND(I)=1                                      CASIN
      IF(WPRIOR(I).LT.0.OR.WPRIOR(I).GT.4)                              CASIN
     1  CALL ERRMES(3HCAS,5HCASIN,4,1)                                  CASIN
      IF(WPRIOR(I).NE.0) NBW=NBW+1                                      CASIN
   20 CONTINUE                                                          CASIN
      IF(NBW.EQ.0) CALL ERRMES(3HCAS,5HCASIN,5,1)                       CASIN
C                                                                       CASIN
C***********************************************************************CASIN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C    READ IN CONTROL CARDS 2 AND 3                                      CASIN
  200 CONTINUE                                                          CASIN
      READ(INP,6000) (( IPRD(I,J),I=1,3),J=1,7),IARG(1),IARG(2)         CASIN
 6000 FORMAT(7(3I2,1X),25X,A4,I2)                                       CASIN
      WRITE(OUTP,2000)                                                  CASIN
      WRITE(OUTP,7000)(I,I=1,7)                                         CASIN
 7000 FORMAT(3X,7(5HIPRD(,I2,4H)    ))                                  CASIN
      WRITE(OUTP,8000) ((IPRD(I,J),I=1,3),J=1,7),IARG(1),IARG(2)        CASIN
 8000 FORMAT(3X,7(I2,1H/,I2,1H/,I2,3X),3X,A4,I2)                        CASIN
      IF(IARG(1).EQ.3HCAS.AND.IARG(2).EQ.2) GO TO 210                   CASIN
      CALL ERRMES(3HCAS,5HCASIN,1,1)                                    CASIN
  210 CONTINUE                                                          CASIN
      READ(INP,6000) ((IPRD(I,J),I=1,3),J=8,14),IARG(1),IARG(2)         CASIN
      WRITE(OUTP,2000)                                                  CASIN
      WRITE(OUTP,7000) (I,I=8,14)                                       CASIN
      WRITE(OUTP,8000) ((IPRD(I,J),I=1,3),J=8,14),IARG(1),IARG(2)       CASIN
      IF(IARG(1).EQ.3HCAS.AND.IARG(2).EQ.3) GO TO 310                   CASIN
      CALL ERRMES(3HCAS,5HCASIN,1,1)                                    CASIN
      GO TO 900                                                         CASIN
  310 CONTINUE                                                          CASIN
      NPDATE=0                                                          CASIN
      DO 320 I=1,14                                                     CASIN
      IF(IPRD(1,I).EQ.0.AND.IPRD(2,I).EQ.0) GO TO 330                   CASIN
      NPDATE=NPDATE+1                                                   CASIN
      IF(IPRD(1,I).GE.64                                                CASIN
     1   .AND.(IPRD(2,I).GE.1.AND.IPRD(2,I).LE.12)                      CASIN
     1   .AND.(IPRD(3,I).GE.1.AND.IPRD(3,I).LE.31)) GO TO 325           CASIN
      IARG(1)=IPRD(1,I)                                                 CASIN
      IARG(2)=IPRD(2,I)                                                 CASIN
      IARG(3)=IPRD(3,I)                                                 CASIN
      CALL ERRMES(3HCAS,5HCASIN,6,1)                                    CASIN
      GO TO 320                                                         CASIN
  325 CONTINUE                                                          CASIN
      CALL LEPA (FLOAT(IPRD(3,I)),IPRD(2,I),IPRD(1,I),0.,DAYS)          CASIN
      PRDATE(I)=IFIX(DAYS)                                              CASIN
  320 CONTINUE                                                          CASIN
  330 CONTINUE                                                          CASIN
      IF(NPDATE.LE.1) GO TO 340                                         CASIN
      ITEMP=NPDATE-1                                                    CASIN
      DO 350 I=1,ITEMP                                                  CASIN
      IF(PRDATE(I).GE.PRDATE(I+1)) CALL ERRMES(3HCAS,5HCASIN,7,1)       CASIN
  350 CONTINUE                                                          CASIN
  340 CONTINUE                                                          CASIN
      S2MAX= 0.25*AREAPS*AREAPS                                         CASIN
      AREACF=.001                                                       CASIN
      YCF=1.0                                                           CASIN
      PRDCF= 1.0E-4                                                     CASIN
      IF(AUNITS.EQ.1) GO TO 400                                         CASIN
      AREACF=.0002471044                                                CASIN
      YCF=1.4869664                                                     CASIN
      PRDCF=3.6743544E-5                                                CASIN
C                                                                       CASIN
  400 IF ( DISTFF .EQ. 0 )  GO TO 900                                   CASIN
C                                                                       CASIN
C        OPEN CAS DISTRIBUTION FILE.                                    CASIN
      DEFINE FILE 4(506,303,U,IDUM)                                             *NEW
      IF ( RSTART .EQ. 0 )  GO TO 900                                   CASIN   **-1
C        RESTART RUN.  READ AND CHECK HEADER RECORD.                    CASIN
      CALL RANACF (CASDIS,1,CASDSB,LCASD,IXDISF,LIXDIS,1)               CASIN
      NT= ICASD(4)                                                      CASIN
```

```
      ARG(6)= 6HCASDIS                                                  CASIN
      ARG(3)= CASDSB(1)                                                 CASIN
C        SHIFT FILENAME 4 CHARACTERS (24 BITS) TO THE LEFT              CASIN
C        BY MULTIPLYING BY 2**24                                        CASIN
      IARG(4)= ICASD(1)*16777216                                        CASIN
      IF ( CASDSB(1) .NE. 6HCASDIS )  CALL ERRMES (3HLEM,5HCASIN,19,1)  CASIN
      IARG(1)= ICASD(2)                                                 CASIN
      ARG(2)= CASDSB(3)                                                 CASIN
      IF ( IARG(1) .NE. ICSFCD .OR. CASDSB(3) .NE. CUNTRY )             CASIN
     1                                CALL ERRMES (3HLEM,5HCASIN,18,1)  CASIN
      IF ( NT .NE. RSTART )  CALL ERRMES (3HLEM,5HCASIN,21,1)           CASIN
C                                                                       CASIN
  900 RETURN                                                            CASIN
      END                                                               CASIN
```

```
FOR,IS CASINL
      SUBROUTINE CASINL                                                 CASINL
C        PERFORMS INITIALIZATION TASKS FOR EACH PREDICTION POINT        CASINL
C                                                                       CASINL
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                      CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)    CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW    ,IBW    ,WINDOW,IPD    ,IPP    ,PPDATE,NREGS CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC   ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
     4  ,LDS1   ,LDS4   ,LDS7   ,LDS8   ,LDS9   ,LDS10 ,LDS11 ,LDS12 ,LDS13 CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR          CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO        FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP    ,OUTP   ,TACQ   ,LTACQ ,CASDSF,LCASDS                   FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF              FILES
C                                                                       FILES
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                     SSHDTA
      COMMON  /SSHDTA/                                                  SSHDTA
```

```
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                    JULY76
      DIMENSION  SSHDTA(39)                                             JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                    SSHDTA
C                                                                       SSHDTA
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C                                                                       CASINL
C        INITIALIZE FLAGS AND COUNTERS                                  CASINL
      RDSSH= 0                                                          CASINL
      NREGS= 0                                                          CASINL
      NZTOT= 0                                                          CASINL
      NSTRAT= 0                                                         CASINL
      NRYES= NYESSK - 1                                                 CASINL
      NRCAMS= NCAMSK - 1                                                CASINL
C        INITIALIZE REGION, ZONE, AND STRATA POINTERS FOR.CAS           CASINL
C        INTERMEDIATE FILE. ( REGION RECORDS ARE 3-12, ZONE RECORDS ARE CASINL
C        13-62, STRATA RECORDS ARE 63-387 )                             CASINL
      IRREG =  2                                                        CASINL
      IRZONE= 12                                                        CASINL
      IRSTR = 62                                                        CASINL
C                                                                       CASINL
C        INITIALLY POSITION FILES YESOUT, SUBHST, CAMSF AT STARTING     CASINL
C        REGION AND ZONE.                                               CASINL
      REWIND YESOUT                                                     CASINL
      REWIND CAMSF                                                      CASINL
      DO 210 I=1,NYESSK                                                 CASINL
      READ (YESOUT)                                                     CASINL
  210 CONTINUE                                                          CASINL
C                                                                       JULY76
      IF ( NT .GT. NSTART .OR. IPP .GT. 1 )  GO TO 225                  JULY76
      REWIND SUBHST                                                     JULY76
      DO 220 I=1,NSSHSK                                                 CASINL
      READ (SUBHST)                                                     CASINL
  220 CONTINUE                                                          CASINL
      NRSSH= NSSHSK - 1                                                 JULY76
```

```
C                                                  JULY76
 225  DO 230 I=1,NCAMSK                            JULY76
      READ ( CAMSF)                                CASINL
 230  CONTINUE                                     CASINL
C                                                  CASINL
 900  RETURN                                       CASINL
      END                                          CASINL
```

```
FOR,IS CASINT
      SUBROUTINE CASINT                                                   CASINT
C         PERFORMS MISCELLANEOUS INITIALIZATION TASKS FOR THE CAS         CASINT
C         SIMULATOR                                                       CASINT
C            1. SETTING COUNTERS AND FLAGS                                CASINT
C            2. OPENING RANDOM ACCESS FILE (CAS INTERMEDIATE FILE)        CASINT
C            3. DETERMINING HOW MANY RECORDS TO SKIP ON THE YESOUT,       CASINT
C               SUBHST, AND CAMSF FILES TO PROPERLY POSITION THOSE FILES CASINT
C               AT THE STARTING REGION AND ZONE (STARTR AND STARTZ)       CASINT
C                                                                         CASINT
C         ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                    ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                             ARGLST
      DIMENSION  IARG(10)                                                 ARGLST
      EQUIVALENCE  ( IARG,ARG )                                           ARGLST
C                                                                         ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                       CASCM
      COMMON  /CASCM /                                                    CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)    CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT  ,AUNITS,DISTFF,BWIND(4)       CASCM
     3  ,WPRIOR(4)     ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE          CASCM
C                                                                         CASCM
C         DATA BLOCK FOR CAS CUMULATIVE FILE                              CASCUM
C         CAS DATA SETS 14, 15, 16, AND 17                                CASCUM
      COMMON  /CASCUM/                                                    CASCUM
     1   CASCUM(32),   BUFFR(504)                                         CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)            CASCUM
     1  ,DSET17(28)                                                       CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                       CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )              CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                      CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                      CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                      CASCUM
C                                                                         CASCUM
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                    CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS    CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS           CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                         CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13    CASFLG
```

```
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                 CASFLG
C                                                                      CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                            CNTRL
      COMMON  /CNTRL /                                                 CNTRL
     1  PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                  CNTRL
      DOUBLE PRECISION  SEED                                           CNTRL
C                                                                      CNTRL
C        CONSTANT QUANTITIES FOR LEM PROGRAM                           CONST
      COMMON  /CONST /                                                 CONST
     1  NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                      CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                 FILES
     1  SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                      FILES
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)      IXCDSF
      COMMON  /IXCDSF/                                                 IXCDSF
     1  IXCDSF(1),LIXCDS
C                                                                      IXCDSF
C        LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                 LEMCM
     1  TITLE(10)    ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                              LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                 LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                    LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                        LEMCM
C                                                                      LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                         PAGECM
      COMMON  /PAGECM/                                                 PAGECM
```

```
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                            PAGECM
C                                                                          PAGECM
C          STATISTICAL INFORMATION FOR LEM                                 STATS
      COMMON  /STATS /                                                     STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR,                STATS
      EQUIVALENCE  ( NT,ITER )                                             STATS
C                                                                          STATS
C          YIELD DATA FROM YESOUT FILE                                     YESDTA
      COMMON  /YESDTA/                                                     YESDTA
     1   YSTR  ,IZPRDD(6)    ,YSCI(6)       ,VSYCI(6)                      YESDTA
     2  ,RDYES ,NYESPP                                                     YESDTA
      INTEGER  RDYES                                                       YESDTA
C                                                                          YESDTA
C         DEBUGGING PRINT FLAG                                             CASINT
      COMMON  /DEBUGF/  DEBUGF                                             CASINT
C                                                                          CASINT
      DATA  DEBUGF  / 0.0 /                                                CASINT
C                                                                          CASINT
      INTEGER  REGION, ZONE, STRATA                                        CASINT
C                                                                          CASINT
C     LOCAL VARIABLES                                                      CASINT
C          COUN   = COUNTRY ID                                             CASINT
C          REGION = REGION ID                                              CASINT
C          ZONE   = ZONE ID                                                CASINT
C          STRATA = STRATUM ID                                             CASINT
C          ICODE  = ERROR MESSAGE CODE FOR ERRMES                          CASINT
C          FILL   = TEMPORARY CELL USED TO FILL OUT LIST IN READ STMT.     CASINT
C          I      = DO LOOP INDEX                                          CASINT
C                                                                          CASINT
C                                                                          CASINT
C          INITIALIZE FLAGS AND COUNTERS                                   CASINT
      PPFLG= 0                                                             CASINT
      IPP  = 0                                                             CASINT
      NSTTL= 0                                                             CASINT
C                                                                          CASINT
C          CLEAR BUFFER REGION FOR CAS CUMULATIVE FILE                     CASINT
      DO 110 I=1,LCASF                                                     CASINT
      BUFFR(I)= 0.0                                                        CASINT
110   CONTINUE                                                             CASINT
C                                                                          CASINT
C          OPEN CAS INTERMEDIATE FILE (RANDOM ACCESS FILE) -- CASDSF       CASINT
```

```
C                                                                       CASINT
C        PERFORM THE FOLLOWING CHECKS ONLY ON THE FIRST ITERATION       CASINT
      IF ( NT .GT. NSTART )  GO TO 900                                  CASINT
C                                                                       CASINT
      H= HH                                                             CASINT
C                                                                       CASINT
C        SKIP OVER THE HEADER RECORD OF THE YESOUT, SUBHST, AND CAMSF   CASINT
C        FILES.                                                         CASINT
      REWIND YESOUT                                                     CASINT
      REWIND SUBHST                                                     CASINT
      REWIND CAMSF                                                      CASINT
      READ (YESOUT)                                                     CASINT
      READ (SUBHST)                                                     CASINT
      READ (CAMSF)                                                      CASINT
      NYESSK= 1                                                         CASINT
      NSSHSK= 1                                                         CASINT
      NCAMSK= 1                                                         CASINT
      IF ( STARTR .EQ. 0 )  GO TO 900                                   JULY76
C        COUNT THE NO. OF RECORDS TO SKIP ON YESOUT                     CASINT
      ARG(3)= 6HYESOUT                                                  CASINT
 210  READ (YESOUT)  COUN,REGION,ZONE,STRATA,YSTR                       CASINT
     1  , ( IZPRDD(I),YSCI(I),VSYCI(I),I=1,6 )                          CASINT
C                                                                       CASINT
      IF ( COUN .EQ. ENDFIL )  GO TO 800                                CASINT
C                                                                       CASINT
      IF ( REGION - STARTR )  250,240,800                               CASINT
C        REGION = STARTING REGION                                       CASINT
 240  IF ( ZONE - STARTZ )  250,300,820                                 CASINT
 250  NYESSK= NYESSK + 1                                                CASINT
      GO TO 210                                                         CASINT
C                                                                       CASINT
C        COUNT NUMBER OF RECORDS TO SKIP ON SUBHST FILE                 CASINT
 300  ARG(3)= 6HSUBHST                                                  CASINT
 310  READ (SUBHST)  COUN.REGION,ZONE, ( FILL,I=4,LSUBH )               CASINT
      IF ( COUN .EQ. ENDFIL )  GO TO 800                                CASINT
      IF ( REGION - STARTR )  350,340,800                               CASINT
C        REGION = STARTING REGION                                       CASINT
 340  IF ( ZONE - STARTZ )  350,400,820                                 CASINT
 350  NSSHSK= NSSHSK + 1                                                CASINT
      GO TO 310                                                         CASINT
C                                                                       CASINT
```

```
C          COUNT NUMBER OF RECORDS TO SKP ON CAMSF                            CASINT
  400   ARG(3)= 5HCAMSF                                                        CASINT
  410   READ (CAMSF)  COUN,REGION,ZONE, ( FILL,I=4,LCAMSF )                    CASINT
        IF ( COUN .EQ. ENDFIL )  GO TO 800                                     CASINT
        IF ( REGION - STARTR )  450,440,800                                    CASINT
C          REGION = STARTING REGION                                            CASINT
  440   IF ( ZONE - STARTZ )  450,900,820                                      CASINT
  450   NCAMSK= NCAMSK + 1                                                     CASINT
        GO TO 410                                                              CASINT
C                                                                              CASINT
C          ERROR.  STARTING REGION NOT FOUND ON FILE                           CASINT
  800   ICODE= 3                                                               CASINT
        ARG(1)= 6HREGION                                                       CASINT
        GO TO 830                                                              CASINT
C                                                                              CASINT
C          ERROR.  STARTING ZONE NOT FOUND ON FILE                             CASINT
  820   ICODE= 4                                                               CASINT
        ARG(1)= 6HZONE                                                         CASINT
  830   CALL ERRMES (3HCAS,6HCASINT,ICODE,1)                                   CASINT
C                                                                              CASINT
  900   RETURN                                                                 CASINT
        END                                                                    CASINT
```

```
FOR,IS CASOUT
      SUBROUTINE CASOUT(ILEVEL)                                         CASOUT
C     THIS SUBROUTINE PRINTS THE AREA AND PRODUCTION REPORT AND SAVES DATACASOUT
C     FOR THE COUNTRY REPORT                                            CASOUT
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1    NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ         FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO        FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD        FILES
     4   ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                     FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C         SUMMARY DATA FOR REPORTS                                      SUMDTA
      COMMON  /SUMDTA/                                                  SUMDTA
     1    CVAEPT,CVEPTA,SDPER ,CVPEPT,CVEPTP,CSUMR(18,18)               SUMDTA
C                                                                       SUMDTA
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
      COMMON  /CASCM /                                                  CASCM
     1    AREACF,YCF   ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2   ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)     CASCM
     3   ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C         DATA BLOCK FOR CAS CUMULATIVE FILE                            CASCUM
C         CAS DATA SETS 14, 15, 16, AND 17                              CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1    CASCUM(32),    BUFFR(504)                                     CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1   ,DSET17(28)                                                    CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1   , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                   CASCUM
     2   , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                   CASCUM
     3   , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                   CASCUM
```

```
C                                                                       CASCUM
C       FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C       LEM CONTROL CARD INPUT DATA                                     LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C       PAGE EJECT CONTROL PARAMETERS FOR LEM                           PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C       STATISTICAL INFORMATION FOR LEM                                 STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C       CAS DATA SET 7 (AT ZONE LEVEL)                                  DSET7
      COMMON  /DSET7 /                                                  DSET7
     1   ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3  JULY76
     2  ,ESTVZ ,HWAZ12                                                  JULY76
     3  ,M1K2CL(10)    ,EPWCL(10)     ,EPW2CL(10)    ,PKPICL(10)        JULY76
     4  ,PK2CL(10)     ,PKCL(10)      ,SSQ(10)                          JULY76
      INTEGER  ZONE                                                     JULY76
      REAL  M1K2KZ, M1K2CL                                              JULY76
      DIMENSION  DSET7(81)                                              JULY76
```

```
      EQUIVALENCE  ( DSET7,ZONE )                                        DSET7
C                                                                        DSET7
C                                                                        CASOUT
      EQUIVALENCE (CASCUM(1),IREG),(CASCUM(2),IZONE),(CASCUM(3),ISTRA),  CASOUT
     1   (CASCUM(5),HWA),   (CASCUM(6),TWA),   (CASCUM(7),EWA),          CASOUT
     1   (CASCUM(8),AERR),   (CASCUM(9),AVAR), (CASCUM(10),TPROD),       CASOUT
     1   (CASCUM(11),EPROD),   (CASCUM(12),PRERR),   (CASCUM(13),PRVAR), CASOUT
     1   (CASCUM(14),TY),    (CASCUM(15),EY),   (CASCUM(16),YERR),       CASOUT
     1   (CASCUM(22),ANAV),                                              CASOUT
     1   (CASCUM(23),ANPRV), (CASCUM(24),SQAER), (CASCUM(25),SQPER),     CASOUT
     1   (CASCUM(26),SQYER), (CASCUM(27),CLEWA), (CASCUM(28),CLEPRD),    CASOUT
     1   (CASCUM(29),CLATEC), (CASCUM(30),CLPTEC), (CASCUM(31),CLATWC),  CASOUT
     1   (CASCUM(32),CLPTWC)                                             CASOUT
      INTEGER M1J,M2J,CT1,CT2,CT3                                        CASOUT
C                                                                        CASOUT
C                                                                        CASOUT
C    CONVERT UNITS AND GET MEAN ( CASCUM)                                CASOUT
      FNT=FLOAT(NT)                                                      CASOUT
      HWA =HWA*AREACF/FNT                                                CASOUT
      TWA =TWA*AREACF/FNT                                                CASOUT
      EWA =EWA*AREACF/FNT                                                CASOUT
      AERR =AERR*AREACF/FNT                                              CASOUT
      AVAR =AVAR*AREACF*AREACF/FNT                                       CASOUT
      TPROD =TPROD*PRDCF/FNT                                             CASOUT
      EPROD =EPROD*PRDCF/FNT                                             CASOUT
      PRERR =PRERR*PRDCF/FNT                                             CASOUT
      PRVAR =PRVAR*PRDCF*PRDCF/FNT                                       CASOUT
      TY =TY*YCF/FNT                                                     CASOUT
      EY =EY*YCF/FNT                                                     CASOUT
      YERR =YERR*YCF/FNT                                                 CASOUT
      M1J = CASCUM(17)/FNT                                               CASOUT
      M2J = CASCUM(18)/FNT                                               CASOUT
      CT1 = CASCUM(19)/FNT                                               CASOUT
      CT2 = CASCUM(20)/FNT                                               CASOUT
      CT3 = CASCUM(21)/FNT                                               CASOUT
      ANAV =ANAV*AREACF*AREACF/FNT                                       CASOUT
      ANPRV =ANPRV*PRDCF*PRDCF/FNT                                       CASOUT
      SQAER =SQAER*AREACF*AREACF                                         CASOUT
      SQPER =SQPER*PRDCF*PRDCF                                           CASOUT
      SQYER =SQYER*YCF*YCF                                               CASOUT
C                                                                        CASOUT
```

```
C    COMPUTE VALUES FOR COUNTRY (CASCUM)                                 CASOUT
       IF(ILEVEL.NE.0) GO TO 10                                          CASOUT
       CLEWA =CLEWA/FNT                                                  CASOUT
       CLEPRD =CLEPRD/FNT                                                CASOUT
       CLATEC =CLATEC/FNT                                                CASOUT
       CLPTEC=CLPTEC/FNT                                                 CASOUT
       CLATWC =CLATWC/FNT                                                CASOUT
       CLPTWC =CLPTWC/FNT                                                CASOUT
C                                                                        CASOUT
C    COMPUTE OTHER VALUES (SUMDTA )                                      CASOUT
   10 CONTINUE                                                           CASOUT
       CVAEPT= SQRT(AVAR)/TWA*100.0                                      CASOUT
       CVPEPT= SQRT(PRVAR)/TPROD*100.0                                   CASOUT
       CVEPTA=0.                                                         CASOUT
       SDPER=0.                                                          CASOUT
       CVEPTP=0.                                                         CASOUT
       IF(NT.EQ.1) GO TO 20                                              CASOUT
       ARG(1)= ( SQAER - AERR*AERR*FNT )/(FNT-1.0)                       CASOUT
       IF ( ARG(1) .GT. 0.0 )  CVEPTA = ( SQRT(ARG(1))/TWA )*100.0       CASOUT
       ARG(1)= ( SQYER - YERR*YERR*FNT )/(FNT-1.0)                       CASOUT
       IF ( ARG(1) .GT. 0.0 )  SDPER = SQRT( ARG(1) )                    CASOUT
       ARG(1)= ( SQPER - PRERR*PRERR*FNT )/(FNT-1.0)                     CASOUT
       IF ( ARG(1) .GT. 0.0 )  CVEPTP = SQRT( ARG(1) )/TPROD *100.0      CASOUT
   20 CONTINUE                                                           CASOUT
C                                                                        CASOUT
C*****************************************************                   CASOUT
C PRINT LINE ON AREA REPORT                                              CASOUT
        IF(APREP.EQ.0) GO TO 30                                          CASOUT
C                                                                        CASOUT
C    NEW PAGE                                                            CASOUT
        IF(ILEVEL .NE.1) GO TO 50                                        CASOUT
       IF ( ENDREG .NE. 0 )  GO TO 25                                    CASOUT
       ICHK= NSTRAZ +9 +NLINE                                            CASOUT
       IF( ICHK.LT.MXLINE) GO TO 50                                      CASOUT
 25    CALL APHDR                                                        CASOUT
C                                                                        CASOUT
C    STRATA LINE                                                         CASOUT
   50 CONTINUE                                                           CASOUT
       IF(ILEVEL.LE.0) GO TO 60                                          CASOUT
       CALL PAGER(1)                                                     CASOUT
       WRITE (OUTP,1000) IREG,IZONE,ISTRA,                               CASOUT
```

```
     1  TWA,EWA,CT1,CT2,CT3,M1J,M2J,CVAEPT,CVEPTA,TY,EY,                   CASOUT
     1  SDPER,TPROD,EPROD,CVPEPT,CVEPTP                                     CASOUT
 1000 FORMAT (1X,2(I3,1X),I4,2X,2(F8.1,1X),3(1X,I4),2X,                     CASOUT
     1 2(I4,1X),1X,2(F6.1,1X),1X,3(F6.2,1X),1X,2(F8.1,1X),2(F6.1,1X))       CASOUT
      GO TO 30                                                              CASOUT
C                                                                           CASOUT
C    ZONE LINE                                                              CASOUT
   60 CONTINUE                                                              CASOUT
      IF(ILEVEL.NE.-2) GO TO 70                                             CASOUT
      CALL PAGER(6)                                                         CASOUT
      WRITE (OUTP,3000)                                                     CASOUT
      WRITE(OUTP,2000)                                                      CASOUT
 2000 FORMAT(1X,20(6H******))                                               CASOUT
      WRITE(OUTP,3000)                                                      CASOUT
 3000 FORMAT(2X)                                                            CASOUT
      WRITE(OUTP,4000)                                                      CASOUT
 4000 FORMAT(1X,3HREG,2X,4HZONE)                                            CASOUT
      WRITE (OUTP,5000) IREG,IZONE,                                         CASOUT
     1  TWA,EWA,CT1,CT2,CT3,M1J,M2J,CVAEPT,CVEPTA,TY,EY,                   CASOUT
     1  SDPER,TPROD,EPROD,CVPEPT,CVEPTP                                     CASOUT
 5000 FORMAT (1X,2(I3,1X),6X,    2(F8.1,1X),3(1X,I4),2X,                    CASOUT
     1 2(I4,1X),1X,2(F6.1,1X),1X,3(F6.2,1X),1X,2(F8.1,1X),2(F6.1,1X))       CASOUT
      WRITE(OUTP,3000)                                                      CASOUT
      WRITE(OUTP,2000)                                                      CASOUT
      WRITE(OUTP,2000)                                                      CASOUT
      WRITE(OUTP,3000)                                                      CASOUT
      GO TO 30                                                              CASOUT
C                                                                           CASOUT
C    REGION LEVEL                                                           CASOUT
   70 CONTINUE                                                              CASOUT
      IF(ILEVEL.NE.-1) GO TO 80                                             CASOUT
      CALL PAGER(4)                                                         CASOUT
      WRITE (OUTP,6000) IREG,                                               CASOUT
     1  TWA,EWA,CT1,CT2,CT3,M1J,M2J,CVAEPT,CVEPTA,TY,EY,                   CASOUT
     1  SDPER,TPROD,EPROD,CVPEPT,CVEPTP                                     CASOUT
 6000 FORMAT (1X,6HREGION,3X,I3,2X,2(F8.1,1X),3(1X,I4),2X,                  CASOUT
     1 2(I4,1X),1X,2(F6.1,1X),1X,3(F6.2,1X),1X,2(F8.1,1X),2(F6.1,1X))       CASOUT
      GO TO 30                                                              CASOUT
C                                                                           CASOUT
C    COUNTRY LEVEL                                                          CASOUT
   80 CONTINUE                                                              CASOUT
```

```
      CALL PAGER(5)                                                     CASOUT
      WRITE (OUTP,3000)                                                 CASOUT
      WRITE(OUTP,2000)                                                  CASOUT
      WRITE(OUTP,2000)                                                  CASOUT
      WRITE(OUTP,3000)                                                  CASOUT
      WRITE (OUTP,7000)                                                 CASOUT
     1  TWA,EWA,CT1,CT2,CT3,M1J,M2J,CVAEPT,CVEPTA,TY,EY,                CASOUT
     1  SDPER,TPROD,EPROD,CVPEPT,CVEPTP                                 CASOUT
 7000 FORMAT (1X,7HCOUNTRY,7X,     2(F8.1,1X),3(1X,I4),2X,              CASOUT
     1 2(I4,1X),1X,2(F6.1,1X),1X,3(F6.2,1X),1X,2(F8.1,1X),2(F6.1,1X))   CASOUT
C                                                                       CASOUT
C*********************************************************              CASOUT
C    SAVE VALUES FOR COUNTRY REPORT                                     CASOUT
   30 CONTINUE                                                          CASOUT
      IF(ILEVEL.NE.0) RETURN                                            CASOUT
      CSUMR(1,IPP)=EWA                                                  CASOUT
      CSUMR(2,IPP)= (SQRT(ANAV)/TWA)*100.                               CASOUT
      CSUMR(3,IPP)=CVAEPT                                               CASOUT
      CSUMR(4,IPP)=CVEPTA                                               CASOUT
      CSUMR(5,IPP)=EY                                                   CASOUT
      CSUMR(6,IPP)=SDPER                                                CASOUT
      CSUMR(7,IPP)=EPROD                                                CASOUT
      CSUMR(8,IPP)=(SQRT(ANPRV)/TPROD)*100.                             CASOUT
      CSUMR(9,IPP)=CVPEPT                                               CASOUT
      CSUMR(10,IPP)=CVEPTP                                              CASOUT
      CSUMR(12,IPP)=CLEWA                                               CASOUT
      CSUMR(13,IPP)=CLATEC                                              CASOUT
      CSUMR(14,IPP)=CLATWC                                              CASOUT
      CSUMR(16,IPP)=CLEPRD                                              CASOUT
      CSUMR(17,IPP)=CLPTEC                                              CASOUT
      CSUMR(18,IPP)=CLPTWC                                              CASOUT
  900 RETURN                                                            CASOUT
      END                                                               CASOUT
```

```
 FOR,IS CASPP
      SUBROUTINE CASPP                                                   CASPP
C        PERFORMS THE FIRST PASS CAS COMPUTATIONS GENERATING DATA SETS  CASPP
C        1-9. CASPP ALSO CALLS CAS3 TO GENERATE DATA SETS 10-19.        CASPP
C                                                                       CASPP
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CONSTANT QUANTITIES FOR LEM PROGRAM                            CONST
      COMMON  /CONST /                                                  CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C        CAS DATA SETS 1,2, AND 3                                       DSET1
      COMMON  /DSET1 /                                                  DSET1
     1   ISUBST,TWAK  ,HWAK  ,FWAK  ,M1K   ,CT1K  ,ANALVK,EPWK  ,EPW2K  JULY76
     2  ,SMPKPI,SUMPK2,SUMPK ,KSUB  ,NCLASS                             JULY76
      REAL  M1K , M2K                                                   JULY76
      DIMENSION  DSET1(14), DSET2(14), DSET3(6)                         JULY76
      EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                         DSET1
     1  , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                               DSET1
C                                                                       DSET1
C        CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                    DSET4
      COMMON  /DSET4 /                                                  DSET4
```

```
     1 . STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1               JULY76
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                   JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS    ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS   ,ANVARS            JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                  JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                         JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )  DSET4
C                                                                      DSET4
C        CAS DATA SET 7 (AT ZONE LEVEL)                                DSET7
      COMMON  /DSET7 /                                                 DSET7
     1   ZONE   ,HWAZ2 ,EZ     ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3 JULY76
     2  ,ESTVZ ,HWAZ12                                                 JULY76
     3  ,M1K2CL(10)    ,EPWCL(10)     ,EPW2CL(10)    ,PKPICL(10)       JULY76
     4  ,PK2CL(10)     ,PKCL(10)      ,SSQ(10)                         JULY76
      INTEGER  ZONE                                                    JULY76
      REAL  M1K2KZ, M1K2CL                                             JULY76
      DIMENSION  DSET7(81)                                             JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                      DSET7
C                                                                      DSET7
C        CAS DATA SET 8 (AT REGION LEVEL)                              DSET8
      COMMON  /DSET8 /                                                 DSET8
     1   REGION,HWAR2 ,ER     ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR JULY76
     2  ,M1M2ZR,FILL8(71)                                              JULY76
      INTEGER  REGION                                                  JULY76
      REAL  M1K2KR                                                     JULY76
      DIMENSION  DSET8(10)                                             JULY76
      EQUIVALENCE  ( DSET8,REGION )                                    DSET8
C                                                                      DSET8
C        CAS DATA SET 9 (AT COUNTRY LEVEL)                             DSET9
      COMMON  /DSET9 /                                                 DSET9
     1   COUNTR,HWAC2 ,EC     ,M1K2KC,ANALVC,M1M2ZC,HWAC1 ,EWAC1 ,ESTVC JULY76
      INTEGER  COUNTR                                                  JULY76
      REAL  M1K2KC                                                     JULY76
      DIMENSION  DSET9(9)                                              JULY76
      EQUIVALENCE  ( DSET9,COUNTR )                                    DSET9
C                                                                      DSET9
C        CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                     JULY76
      COMMON  /DSET11/                                                 DSET11
     1   HWAZ   ,TWAZ   ,EWAZ   ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ    ,EYZ    ,YERRZ ,M1Z   ,M2Z    ,CT1Z  ,CT2Z  ,CT3Z  ,ANAVZ  DSET11
```

```
     3   ,ANPRVZ                                                          DSET11
      REAL  M1Z , M2Z                                                     DSET11
      DIMENSION  DSET11(19)                                               DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                        DSET11
C                                                                         DSET11
C        CAS DATA SET 12 (REGION DATA -- FINAL PASS)                      JULY76
      COMMON  /DSET12/                                                    DSET12
     1    HWAR   ,TWAR   ,EWAR   ,AERRR  ,AVARR  ,TPRODR,EPRODR,PRERRR,PRVARR DSET12
     2   ,TYR    ,EYR    ,YERRR  ,M1R    ,M2R    ,CT1R   ,CT2R   ,CT3R   ,ANAVR  DSET12
     3   ,ANPRVR                                                          DSET12
      REAL  M1R , M2R                                                     DSET12
      DIMENSION  DSET12(19)                                               DSET12
      EQUIVALENCE  ( DSET12,HWAR )                                        DSET12
C                                                                         DSET12
C        CAS DATA SET 13 (COUNTRY DATA -- FINAL PASS)                     JULY76
      COMMON  /DSET13/                                                    DSET13
     1    HWAC   ,TWAC   ,EWAC   ,AERRC  ,AVARC  ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2   ,TYC    ,EYC    ,YERRC  ,M1C    ,M2C    ,CT1C   ,CT2C   ,CT3C   ,ANAVC  DSET13
     3   ,ANPRVC,CLEWA  ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC               DSET13
      REAL  M1C , M2C                                                     DSET13
      DIMENSION  DSET13(25)                                               DSET13
      EQUIVALENCE  ( DSET13,HWAC )                                        DSET13
C                                                                         DSET13
C        FILE DEFINITIONS AND RECORD LENGTHS                              FILES
      COMMON  /FILES /                                                    FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO         FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
     4   ,INP    ,OUTP   ,TACQ   ,LTACQ ,CASDSF,LCASDS                    FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT   FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF                FILES
C                                                                         FILES
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)         IXCDSF
      COMMON  /IXCDSF/                                                    IXCDSF
     1    IXCDSF(1),LIXCDS
C                                                                         IXCDSF
C        INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE     IXSUBH
      COMMON  /IXSUBH/                                                    IXSUBH
     1 LIXSSH,IXSUBH(1)                                                   MOD1
C                                                                         IXSUBH
      COMMON/FILES1/                                                      FILES1
```

```
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C         LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISFXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C         SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                    SSHDTA
      COMMON  /SSHDTA/                                                  SSHDTA
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                    JULY76
      DIMENSION  SSHDTA(39)                                             JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                    SSHDTA
C                                                                       SSHDTA
C         STATISTICAL INFORMATION FOR LEM                               STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C        DEBUGGING PRINT FLAG                                           CASPP
      COMMON  /DEBUGF/  DEBUGF                                          CASPP
C                                                                       CASPP
C         YIELD DATA FROM YESOUT FILE                                   YESDTA
      COMMON  /YESDTA/                                                  YESDTA
     1   YSTR  ,IZPROD(6)     ,YSCI(6)       ,VSYCI(6)                  YESDTA
     2  ,RDYES ,NYESPP                                                  YESDTA
      INTEGER  RDYES                                                    YESDTA
C                                                                       YESDTA
C                                                                       CASPP
C         INITIALLY POSITION FILES YESOUT, SUBHST, AND CAMSF AS SPECIFIEDCASPP
C         BY STARTR AND STARTZ. ALSO INITIALIZE FLAGS AND COUNTERS.     CASPP
      IPP= IPP + 1                                                      CASPP
```

```
      CALL CASINL                                                        CASPP
C        ON THE FIRST ITERATION CALL CLASSN FOR EACH PREDICTION POINT    JULY76
C        TO DETERMINE THE CLASS NUMBER FOR EACH SUBSTRATA                JULY76
      IF ( NT .EQ. NSTART )  CALL CLASSN                                 JULY76
      NRSSH = 0                                                          JULY76
C                                                                        CASPP
C        INITIALIZE DATA SET 9 (COUNTRY LEVEL)                           CASPP
      DO 110 I=2,LDS9                                                    CASPP
      DSET9(I)= 0.0                                                      CASPP
  110 CONTINUE                                                           CASPP
      ENDC= 0                                                            CASPP
      TWAC= 0.0                                                          CASPP
C                                                                        CASPP
C        INITIALIZE DATA SET 8 (REGION LEVEL)                            CASPP
  120 DO 130 I=1,LDS8                                                    CASPP
      DSET8(I)= 0.0                                                      CASPP
  130 CONTINUE                                                           CASPP
      ENDREG= 0                                                          CASPP
      TWAR= 0.0                                                          CASPP
C                                                                        CASPP
C        INITIALIZE DATA SET 7 (ZONE LEVEL)                              CASPP
  140 DO 150 I=1,LDS7                                                    CASPP
      DSET7(I)= 0.0                                                      CASPP
  150 CONTINUE                                                           CASPP
      ENDZON= 0                                                          CASPP
      TWAZ= 0.0                                                          CASPP
C                                                                        CASPP
C        INITIALIZE DATA SETS 4,5, AND 6 (STRATA LEVEL)                  CASPP
  160 DO 170 I=1,LDS4                                                    CASPP
      DSET4(I)= 0.0                                                      CASPP
  170 CONTINUE                                                           CASPP
      KSUB= 0                                                            CASPP
C                                                                        CASPP
C        READ STRATA YIELD DATA FROM YESOUT AND SELECT THE PROPER VALUE  CASPP
C        OF ESTIMATED YIELD FOR THE CURRENT BIOWINDOW OR PREDICTION DATECASPP
      CALL GETYS                                                         CASPP
      IF ( NFATAL .NE. 0 )  GO TO 990                                    CASPP
      IF ( YSTR .LT. 0.0 )  GO TO 550                                    CASPP
C                                                                        CASPP
C        SHOULD WE READ SUBHST THIS TIME OR IS SUBSTRATA DATA ALREADY    CASPP
C        READ IN AND WAITING FOR PROCESSING                              CASPP
```

```
      IF ( RDSSH .EQ. 0 )  GO TO 210                                    CASPP
C        SUBSTRATA DATA ALREADY READ IN. SET RDSSH FLAG TO READ SUBHST  CASPP
C        NEXT TIME.                                                     CASPP
      RDSSH= 0                                                          CASPP
      GO TO 250                                                         CASPP
C                                                                       CASPP
C        READ NEXT SUBSTRATA RECORD FROM ISUBH2 FILE                    JULY76
  210 NRSSH= NRSSH + 1                                                  JULY76
      CALL RANACF (ISUBH2,NRSSH,SSHDTA,LSUBH2,IXSUBH,LIXSSH,1)          JULY76
      NCLASS= CLASS(IPP)                                                JULY76
C                                                                       CASPP
C                                                                       CASPP
C           TEMPORARY DEBUGGING PRINTOUT                                CASPP
C                                                                       CASPP
C                                                                       CASPP
C                                                                       CASPP
C        TEST FOR END OF COUNTRY ON ISUBH2 FILE                         JULY76
      IF ( COUN2 .EQ. ENDFIL )  GO TO 400                               CASPP
C        CHECK FOR NEW REGION, ZONE, OR SUBSTRATA                       CASPP
      IF ( IREG2 .NE. REGION )  GO TO 430                               CASPP
      IF ( IZONE2 .NE. ZONE )   GO TO 440                               CASPP
      IF ( ISTRA2 .NE. STRATA)  GO TO 450                               CASPP
C                                                                       CASPP
C        SKIP OVER  GRPNO, NAGR, NA, AND HISTPW CHECKS IF NOT FIRST     CASPP
C        ITERATION FOR THIS RUN.                                        CASPP
  250 IF ( NT .NE. NSTART )  GO TO 290                                  CASPP
      IF ( GRPNO .GT. 0 .AND. GRPNO .LT. 4 )  GO TO 260                 CASPP
C        ILLEGAL GROUP NUMBER  (MUST BE 1,2, OR 3)                      CASPP
      CALL ERRMES (3HCAS,5HCASPP,8,1)                                   CASPP
      GO TO 990                                                         CASPP
C                                                                       CASPP
  260 IF ( NAGR .EQ. 0 )  GO TO 270                                     CASPP
      IF ( NA .NE. 0 )  GO TO 280                                       CASPP
C                                                                       CASPP
C        NAGR= 0 OR NA= 0. PRINT WARNING AND SET GROUP NO. = 3          CASPP
  270 CALL ERRMES (3HCAS,5HCASPP,9,0)                                   CASPP
      GRPNO= 3                                                          CASPP
C                                                                       CASPP
  280 IF ( GRPNO .EQ. 3 )  GO TO 290                                    CASPP
      IF ( HISTPW .GT. 0.0 )  GO TO 290                                 CASPP
C        ERROR.  HISTPW .LE. 0.0 AND GRPNO = 1 OR 2                     CASPP
```

```
      CALL ERRMES (3HCAS,5HCASPP,15,0)                                  CASPP
      GRPNO= 3                                                          CASPP
C                                                                       CASPP
C        GENERATE DATA SETS 1,2, AND 3 AT THE SUBSTRATA LEVEL.          CASPP
  290 CALL DS123                                                        CASPP
      IF ( NFATAL .NE. 0 )  GO TO 990                                   CASPP
C                                                                       CASPP
      GO TO 210                                                         CASPP
C                                                                       CASPP
C        END OF DATA ON SUBHST  (COUNTRY = 4HZZZZ)                      CASPP
C        SET END OF COUNTRY FLAG                                        CASPP
  400 ENDC = 1                                                          JULY76
C        SET END OF REGION FLAG                                         CASPP
  430 ENDREG= 1                                                         CASPP
C        SET END OF ZONE FLAG                                           CASPP
  440 ENDZON= 1                                                         CASPP
C        END OF STRATA.  SET RDSSH TO SKIP READING SUBHST NEXT TIME     CASPP
  450 RDSSH= 1                                                          CASPP
C                                                                       CASPP
C        FINISH PROCESSING DATA SETS 4, 5, AND 6.                       CASPP
      CALL DS456                                                        CASPP
      IF ( NFATAL .NE. 0 )  GO TO 990                                   CASPP
C                                                                       CASPP
  550 IF ( ENDZON .EQ. 0 )  GO TO 160                                   CASPP
C                                                                       CASPP
C        END OF ZONE                                                    CASPP
C        FINISH PROCESSING DATA SET 7 (ZONE LEVEL)                      CASPP
      CALL DS7                                                          CASPP
      IF ( NFATAL .NE. 0 )  GO TO 990                                   CASPP
C                                                                       CASPP
      IF ( ENDREG .EQ. 0 )  GO TO 140                                   CASPP
C                                                                       CASPP
C        END OF REGION                                                  CASPP
C        GENERATE REST OF DATA SET 8 (EQ. 77 -- REGION LEVEL)           CASPP
      IF ( HWAR1 .NE. 0.0 )  ER= EWAR1/HWAR1                            CASPP
C                                                                       CASPP
C        WRITE DATA SET 8 ONTO INTERMEDIATE FILE                        CASPP
      IRREG = IRREG + 1                                                 CASPP
      NREGS = NREGS + 1                                                 CASPP
      CALL RANACF (CASDSF,IRREG,DSET8,LCASDS,IXCDSF,LIXCDS,2)           CASPP
C                                                                       CASPP
```

```
C         AGGREGATE REGION DATA SET 8 UP TO DATA SET 9 (COUNTRY LEVEL)   CASPP
C         EQNS. 80-86,88,89                                               CASPP
      TWAC= TWAC + TWAR                                                   CASPP
      DO 820 I=2,5                                                        JULY76
      DSET9(I)= DSET9(I) + DSET8(I)                                       CASPP
  820 CONTINUE                                                            CASPP
      M1M2ZC= M1M2ZC + M1M2ZR                                             CASPP
      HWAC1 = HWAC1  + HWAR1                                              CASPP
      EWAC1 = EWAC1  + EWAR1                                              CASPP
      IF ( ENDC .EQ. 0 )  GO TO 120                                       CASPP
C                                                                         CASPP
C         END OF COUNTRY                                                  CASPP
C         GENERATE REST OF DATA SET 9 (EQ. 87 -- COUNTRY LEVEL)           CASPP
      IF ( HWAC1 .NE. 0.0)  EC= EWAC1/HWAC1                               CASPP
C                                                                         CASPP
C                                                                         JULY76
C         COMPUTE ESTIMATED GROUP 1,2 VARIANCE OF ALL STRATA WITH         JULY76
C         ACQUIRED SEGMENTS                                               JULY76
      CALL CAS2                                                           JULY76
C                                                                         JULY76
C         GENERATE DATA SET 10-19 ON FINAL PASS FOR EACH PREDICTION POINTJULY76
      CALL CAS3                                                           JULY76
  990 RETURN                                                              CASPP
      END                                                                 CASPP
```

```
FOR,IS CAS2
      SUBROUTINE CAS2                                                    CAS2
C        COMPUTES THE AREA VARIANCE AND ANALYTIC AREA VARIANCE OF        JULY76
C        ALL STRATA WITH ACQUIRED SEGMENTS AND ACCUMULATES QUANTITIES    JULY76
C        AT THE ZONE, REGION, AND COUNTRY LEVELS WHICH WILL BE USED      JULY76
C        TO COMPUTE THE VARIANCE OF STRATA WITHOUT ACQUIRED SEGMENTS     JULY76
C        AND ALSO THE AREA VARIANCES AT THE ZONE, REGION, AND            JULY76
C        COUNTRY LEVELS.                                                 JULY76
C                                                                        CAS2
C        ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                       CASCM
      COMMON  /CASCM /                                                   CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)   ,PPRUTS(5,2)   ,YPRUTS(3,2) CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)     CASCM
     3  ,WPRIOR(4)     ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE         CASCM
C                                                                        CASCM
C        DATA BLOCK FOR CAS CUMULATIVE FILE                              CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                                CASCUM
      COMMON  /CASCUM/                                                   CASCUM
     1   CASCUM(32),    BUFFR(504)                                       CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)           CASCUM
     1  ,DSET17(28)                                                      CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                      CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )             CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                     CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                     CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                     CASCUM
C                                                                        CASCUM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                   CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT  ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC   ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1   ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR          CASFLG
```

```
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CONSTANT QUANTITIES FOR LEM PROGRAM                            CONST
      COMMON  /CONST /                                                  CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
      COMMON/FILES1/                                                    FILES1
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C        CAS DATA SETS 1,2, AND 3                                       DSET1
      COMMON  /DSET1 /                                                  DSET1
     1   ISUBST,TWAK  ,HWAK  ,EWAK  ,M1K   ,CT1K  ,ANALVK,EPWK  ,EPW2K  JULY76
     2  ,SMPKPI,SUMPK2,SUMPK ,KSUB  ,NCLASS                             JULY76
      REAL  M1K , M2K                                                   JULY76
      DIMENSION  DSET1(14), DSET2(14), DSET3(6)                         JULY76
      EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                         DSET1
     1  , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                               DSET1
C                                                                       DSET1
C        CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                    DSET4
      COMMON  /DSET4 /                                                  DSET4
     1   STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                JULY76
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                    JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS   ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS  ,ANVARS               JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                   JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                          JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )   DSET4
C                                                                       DSET4
C        CAS DATA SET 7 (AT ZONE LEVEL)                                 DSET7
      COMMON  /DSET7 /                                                  DSET7
     1   ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3  JULY76
     2  ,ESTVZ ,HWAZ12                                                  JULY76
     3  ,M1K2CL(10)   ,EPWCL(10)    ,EPW2CL(10)   ,PKPICL(10)           JULY76
     4  ,PK2CL(10)    ,PKCL(10)     ,SSQ(10)                            JULY76
      INTEGER  ZONE                                                     JULY76
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

28234-6029-RU-51
Page 416a

```
      REAL  M1K2KZ, M1K2CL                                              JULY76
      DIMENSION  DSET7(81)                                              JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                       DSET7
C                                                                       DSET7
C        CAS DATA SET 8 (AT REGION LEVEL)                               DSET8
      COMMON  /DSET8 /                                                  DSET8
     1   REGION,HWAR2 ,ER     ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR JULY76
     2  ,M1M2ZR,FILL8(71)                                               JULY76
      INTEGER  REGION                                                   JULY76
      REAL  M1K2KR                                                      JULY76
      DIMENSION  DSET8(10)                                              JULY76
      EQUIVALENCE  ( DSET8,REGION )                                     DSET8
C                                                                       DSET8
C        CAS DATA SET 9 (AT COUNTRY LEVEL)                              DSET9
      COMMON  /DSET9 /                                                  DSET9
     1   COUNTR,HWAC2 ,EC     ,M1K2KC,ANALVC,M1M2ZC,HWAC1 ,EWAC1 ,ESTVC JULY76
      INTEGER  COUNTR                                                   JULY76
      REAL  M1K2KC                                                      JULY76
      DIMENSION  DSET9(9)                                               JULY76
      EQUIVALENCE  ( DSET9,COUNTR )                                     DSET9
C                                                                       DSET9
C        CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                      JULY76
      COMMON  /DSET11/                                                  DSET11
     1   HWAZ  ,TWAZ  ,EWAZ  ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ   ,EYZ   ,YERRZ ,M1Z   ,M2Z   ,CT1Z  ,CT2Z  ,CT3Z  ,ANAVZ  DSET11
     3  ,ANPRVZ                                                         DSET11
      REAL  M1Z , M2Z                                                   DSET11
      DIMENSION  DSET11(19)                                             DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                      DSET11
C                                                                       DSET11
C        CAS DATA SET 12 (REGION DATA -- FINAL PASS)                    JULY76
      COMMON  /DSET12/                                                  DSET12
     1   HWAR  ,TWAR  ,EWAR  ,AERRR ,AVARR ,TPRODR,EPRODR,PRERRR,PRVARR DSET12
     2  ,TYR   ,EYR   ,YERRR ,M1R   ,M2R   ,CT1R  ,CT2R  ,CT3R  ,ANAVR  DSET12
     3  ,ANPRVR                                                         DSET12
      REAL  M1R , M2R                                                   DSET12
      DIMENSION  DSET12(19)                                             DSET12
      EQUIVALENCE  ( DSET12,HWAR )                                      DSET12
C                                                                       DSET12
C        CAS DATA SET 13 (COUNTRY DATA -- FINAL PASS)                   JULY76
      COMMON  /DSET13/                                                  DSET13
```

```
     1   HWAC  ,TWAC  ,EWAC  ,AERRC ,AVARC ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2  ,TYC   ,EYC   ,YERRC ,M1C   ,M2C   ,CT1C  ,CT2C  ,CT3C  ,ANAVC  DSET13
     3  ,ANPRVC,CLEWA ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC               DSET13
      REAL  M1C , M2C                                                  DSET13
      DIMENSION  DSET13(25)                                            DSET13
      EQUIVALENCE  ( DSET13,HWAC )                                     DSET13
C                                                                      DSET13
C        FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                 FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF               FILES
C                                                                      FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                   IXCASF
      COMMON  /IXCASF/                                                 IXCASF
     1  IXCASF(1),LIXCAS
C                                                                      IXCASF
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)      IXCDSF
      COMMON  /IXCDSF/                                                 IXCDSF
     1   IXCDSF(1),LIXCDS
C                                                                      IXCDSF
C        INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE  IXSUBH
      COMMON  /IXSUBH/                                                 IXSUBH
     1 LIXSSH,IXSUBH(1)                                                MOD1
C                                                                      IXSUBH
C        LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                 LEMCM
     1   TITLE(10)    ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSFAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                     LEMCM
      DIMENSION  RSEED(7)                                              LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5      LEMCM
     1  ,RSEED6,RSEED7                                                 LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                    LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                        LEMCM
C                                                                      LEMCM
```

```
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER   ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR             STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C        SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                     SSHDTA
      COMMON  /SSHDTA/                                                  SSHDTA
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                    JULY76
      DIMENSION  SSHDTA(39)                                             JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                    SSHDTA
C                                                                       SSHDTA
C       DEBUGGING PRINT FLAG                                            CAS2
      COMMON  /DEBUGF/  DEBUGF                                          CAS2
C                                                                       CAS2
C                                                                       CAS2
C     LOCAL VARIABLES                                                   CAS2
C        I       DO LOOP INDEX                                          CAS2
C        IREG    REGION INDEX (1,2,....,NREG)                           CAS2
C        IZONE   ZONE INDEX (1,2,....,NZONES)                           CAS2
C                                                                       CAS2
C                                                                       CAS2
      IF ( MIK2KC .NE. 0.0 )  GO TO 110                                 CAS2
C                                                                       CAS2
      CALL PAGER (3)                                                    CAS2
      WRITE (OUTP,1)                                                    CAS2
 1    FORMAT (//28H  NO ACQUISITIONS IN COUNTRY)                        CAS2
      IF ( PPFLG .NE. 0 )  GO TO 105                                    CAS2
      WRITE (OUTP,2)  IBW                                               CAS2
 2    FORMAT (16H  FOR BIOWINDOW ,I2)                                   CAS2
      GO TO 990                                                         CAS2
 105  WRITE (OUTP,3)  IPRD(2,IPD),IPRD(3,IPD),IPRD(1,IPD)               CAS2
 3    FORMAT (22H  FOR PREDICTION DATE ,I2,1H/I2,1H/I2)                 CAS2
      GO TO 990                                                         CAS2
C                                                                       CAS2
```

```
C         INITIALIZE REGION, ZONE, AND STRATA POINTERS FOR CAS          CAS2
C         INTERMEDIATE FILE. ( REGION RECORDS ARE 3-12, ZONE RECORDS ARE CAS2
C         13-62, STRATA RECORDS ARE 63-387 )                            CAS2
  110 IRREG= 2                                                          CAS2
      IRZONE= 12                                                        CAS2
      IRSTR = 62                                                        CAS2
      IREG= 0                                                           CAS2
      NRSSH = 0                                                         JULY76
C                                                                       JULY76
C         REGION LOOP                                                   JULY76
  140 IREG = IREG + 1                                                   JULY76
      IZONE= 0                                                          CAS2
C                                                                       CAS2
C         READ DATA SET 8 (REGION LEVEL) FROM CAS INTERMEDIATE FILE     CAS2
      IRREG= IRREG + 1                                                  CAS2
      CALL RANACF (CASDSF,IRREG,DSET8,LCASDS,IXCDSF,LIXCDS,1)           CAS2
C                                                                       CAS2
C         ZONE LOOP                                                     JULY76
  180 IZONE = IZONE + 1                                                 CAS2
C                                                                       CAS2
C         READ DATA SET 7 (ZONE LEVEL) FROM CAS INTERMEDIATE FILE       CAS2
      IRZONE= IRZONE + 1                                                CAS2
      CALL RANACF (CASDSF,IRZONE,DSET7,LCASDS,IXCDSF,LIXCDS,1)          CAS2
C                                                                       CAS2
      ISTRAZ = 0                                                        JULY76
C                                                                       JULY76
C         STRATA LOOP                                                   JULY76
  200 ISTRAZ = ISTRAZ + 1                                               JULY76
C         READ DATA SETS 4,5, AND 6 FROM CAS INTERMEDIATE FILE          JULY76
      IRSTR = IRSTR + 1                                                 JULY76
      CALL RANACF (CASDSF,IRSTR,DSET4,LCASDS,IXCDSF,LIXCDS,1)           JULY76
C                                                                       JULY76
      V1V2S = 0.0                                                       JULY76
      NSUB = XCT1S + XCT2S + XCT3S + 0.01                               JULY76
      ISUB = 0                                                          JULY76
C                                                                       JULY76
C         SUBSTRATA LOOP                                                JULY76
C         READ SUBSTRATA DATA FROM ISUBH2 FILE                          JULY76
  220 ISUB = ISUB + 1                                                   JULY76
      NRSSH = NRSSH + 1                                                 JULY76
      CALL RANACF (ISUBH2,NRSSH,SSHDTA,LSUBH2,IXSUBH,LIXSSH,1)          JULY76
```

```
C                                                                       JULY76
      NCLASS= CLASS(IPP)                                                JULY76
C                                                                       JULY76
C                                                                       JULY76
C                                                                       JULY76
C                                                                       JULY76
C         IF CLASS NUMBER IS ZERO OR IF NO ACQUIRED SEGMENTS IN STRATA, JULY76
C         SKIP THIS SUBSTRATUM.                                         JULY76
      IF ( NCLASS .EQ. 0 )  GO TO 250                                   JULY76
      IF ( M1K2KZ .LT. 2.0 )  GO TO 250                                 JULY76
      IF ( XM1JS + XM2JS .EQ. 0.0 )  GO TO 250                          JULY76
C                                                                       JULY76
      IF ( GRPNO - 2 )  240,230,250                                     JULY76
C         GROUP 2 SUBSTRATA. FINISH COMPUTING GROUP 2 VARIANCE MULTIPLIERJULY76
 230  IF ( XM2JS .EQ. 0.0 )  GO TO 250                                  JULY76
      VMULTK = VMULTK*HWAS2/XM2JS                                       JULY76
C         GROUP 1  OR GROUP 2 SUBSTRATA                                 JULY76
 240  V1V2S = V1V2S + VMULTK*SSQ(NCLASS)                                JULY76
C         TEST FOR END OF STRATUM                                       JULY76
 250  IF ( ISUB .LT. NSUB )  GO TO 220                                  JULY76
C                                                                       JULY76
C         IF NO ACQUIRED SEGMENTS IN STRATUM OR IF LESS THAN 2 SEGMENTS JULY76
C         IN ZONE, SKIP VARIANCE CALCULATIONS FOR STRATUM.              JULY76
C         (COMPUTED LATER IN SUBROUTINE DS10)                           JULY76
      IF ( M1K2KZ .LT. 2.0 )  GO TO 260                                 JULY76
      IF ( XM1JS + XM2JS .EQ. 0.0 )  GO TO 260                          JULY76
C         COMPUTE AREA VARIANCE FOR STRATA WITH ACQUIRED SEGMENTS.      JULY76
      V1V2S = V1V2S + T                                                 JULY76
      ANVS2 = ANVS2 + T                                                 JULY76
      TAU2S = ( 1.0 + HWAS3/(HWAS1 + HWAS2) )**2                        JULY76
      VARS = TAU2S*V1V2S                                                JULY76
      ANVARS= TAU2S*(ANVS1 + ANVS2)                                     JULY76
C                                                                       JULY76
C         WRITE STRATA RECORD BACK ONTO CASDSF                          JULY76
      CALL KANACF (CASDSF,IRSTR,DSET4,LCASDS,IXCDSF,LIXCDS,2)           JULY76
      ESTVZ = ESTVZ + V1V2S                                             JULY76
      ANALVZ= ANALVZ + ANVS1 + ANVS2                                    JULY76
      GO TO 265                                                         JULY76
C                                                                       JULY76
C         COMPUTE TOTAL WHEAT AREA FOR STRATA WITHOUT SEGMENTS          JULY76
C         OR FOR STRATA IN A ZONE WITH LESS THAN 2 ACQUIRED SEGMENTS.   JULY76
```

```
  260  HWAZ3 = HWAZ3 + HWAS1 + HWAS2 + HWAS3                                JULY76
 C                                                                           JULY76
   265 CONTINUE                                                              JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C        TEST FOR END OF ZONE                                               JULY76
  270  IF ( ISTRAZ .LT. NSTRAZ )  GO TO 200                                  JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C                                                                           JULY76
  276  CONTINUE                                                              JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C        WRITE DATA SET 7 BACK ONTO CAS INTERMEDIATE FILE (CASDSF)          JULY76
       CALL RANACF (CASDSF,IRZONE,DSET7,LCASDS,IXCDSF,LIXCDS,2)              JULY76
 C                                                                           JULY76
       IF ( M1K2KZ .LT. 2.0 )  GO TO 280                                     JULY76
       ESTVR = ESTVR + ESTVZ                                                 JULY76
       ANALVR= ANALVR + ANALVZ                                               JULY76
 C        TEST FOR END OF REGION                                             JULY76
  280  IF ( IZONE .LT. NZONES )  GO TO 180                                   JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C                                                                           JULY76
       IF ( M1M2ZR .EQ. 0 )  GO TO 290                                       JULY76
 C        WRITE DATA SET 8 BACK ONTO CASDSF                                  JULY76
       CALL RANACF (CASDSF,IRREG,DSET8,LCASDS,IXCDSF,LIXCDS,2)               JULY76
       ESTVC = ESTVC + ESTVR                                                 JULY76
       ANALVC = ANALVC + ANALVR                                              JULY76
 C                                                                           JULY76
 C        TEST FOR END OF COUNTRY                                            JULY76
  290  IF ( IREG .LT. NREGS )  GO TO 140                                     JULY76
 C                                                                           JULY76
 C                                                                           JULY76
 C                                                                           CAS2
  990  RETURN                                                                CAS2
 C                                                                           CAS2
 C                                                                           CAS2
       END                                                                   CAS2
```

```
FOR,IS CAS3
      SUBROUTINE CAS3                                                    CAS3
C        GENERATES DATA SETS 10-17,19 USING DATA SETS 1-9 READ FROM      CAS3
C        THE CAS INTERMEDIATE FILE.                                      CAS3
C                                                                        CAS3
C        ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                       CASCM
      COMMON  /CASCM /                                                   CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)   CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)       CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)            CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE         CASCM
C                                                                        CASCM
C        DATA BLOCK FOR CAS CUMULATIVE FILE                              CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                                CASCUM
      COMMON  /CASCUM/                                                   CASCUM
     1   CASCUM(32),   BUFFR(504)                                        CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)           CASCUM
     1  ,DSET17(28)                                                      CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                      CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )             CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                     CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                     CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                     CASCUM
C                                                                        CASCUM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                   CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS   CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS          CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                        CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13   CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                   CASFLG
C                                                                        CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                              CNTRL
      COMMON  /CNTRL /                                                   CNTRL
```

```
     1    PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                 CNTRL
      DOUBLE PRECISION  SEED                                          CNTRL
C                                                                     CNTRL
C         CAS DATA SET 7 (AT ZONE LEVEL)                              DSET7
      COMMON  /DSET7 /                                                DSET7
     1    ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3  JULY76
     2  ,ESTVZ ,HWAZ12                                                JULY76
     3  ,M1K2CL(10)   ,EPWCL(10)    ,EPW2CL(10)   ,PKPICL(10)        JULY76
     4  ,PK2CL(10)    ,PKCL(10)     ,SSQ(10)                         JULY76
      INTEGER  ZONE                                                   JULY76
      REAL  M1K2KZ, M1K2CL                                            JULY76
      DIMENSION  DSET7(81)                                            JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                     DSET7
C                                                                     DSET7
C         CAS DATA SET 8 (AT REGION LEVEL)                            DSET8
      COMMON  /DSET8 /                                                DSET8
     1    REGION,HWAR2 ,ER    ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR  JULY76
     2  ,M1M2ZR,FILL8(71)                                             JULY76
      INTEGER  REGION                                                 JULY76
      REAL  M1K2KR                                                    JULY76
      DIMENSION  DSET8(10)                                            JULY76
      EQUIVALENCE  ( DSET8,REGION )                                   DSET8
C                                                                     DSET8
C         CAS DATA SET 9 (AT COUNTRY LEVEL)                           DSET9
      COMMON  /DSET9 /                                                DSET9
     1    COUNTR,HWAC2 ,EC    ,M1K2KC,ANALVC,M1M2ZC,HWAC1 ,EWAC1 ,ESTVC  JULY76
      INTEGER  COUNTR                                                 JULY76
      REAL  M1K2KC                                                    JULY76
      DIMENSION  DSET9(9)                                             JULY76
      EQUIVALENCE  ( DSET9,COUNTR )                                   DSET9
C                                                                     DSET9
C         CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                   JULY76
      COMMON  /DSET11/                                                DSET11
     1    HWAZ  ,TWAZ  ,EWAZ  ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ   ,EYZ   ,YERRZ ,M1Z   ,M2Z   ,CT1Z  ,CT2Z  ,CT3Z  ,ANAVZ  DSET11
     3  ,ANPRVZ                                                       DSET11
      REAL  M1Z , M2Z                                                 DSET11
      DIMENSION  DSET11(19)                                           DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                    DSET11
C                                                                     DSET11
```

```
C         CAS DATA SET 12 (REGION DATA -- FINAL PASS)                           JULY76
      COMMON   /DSET12/                                                         DSET12
     1    HWAR   ,TWAR    ,EWAR    ,AERRR  ,AVARR  ,TPRODR,EPRODR,PRERRR,PRVARR DSET12
     2   ,TYR    ,EYR     ,YERRR  ,M1R    ,M2R    ,CT1R   ,CT2R   ,CT3R   ,ANAVR  DSET12
     3   ,ANPRVR                                                                DSET12
      REAL   M1R , M2R                                                          DSET12
      DIMENSION  DSET12(19)                                                     DSET12
      EQUIVALENCE  ( DSET12,HWAR )                                              DSET12
C                                                                               DSET12
C         CAS DATA SET 13 (COUNTRY DATA -- FINAL PASS)                          JULY76
      COMMON   /DSET13/                                                         DSET13
     1    HWAC   ,TWAC    ,EWAC    ,AERRC  ,AVARC  ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2   ,TYC    ,EYC     ,YERRC  ,M1C    ,M2C    ,CT1C   ,CT2C   ,CT3C   ,ANAVC  DSET13
     3   ,ANPRVC,CLEWA ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC                     DSET13
      REAL   M1C , M2C                                                          DSET13
      DIMENSION  DSET13(25)                                                     DSET13
      EQUIVALENCE  ( DSET13,HWAC )                                              DSET13
C                                                                               DSET13
C         FILE DEFINITIONS AND RECORD LENGTHS                                   FILES
      COMMON   /FILES /                                                         FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ                 FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO               FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD               FILES
     4   ,INP    ,OUTP   ,TACQ   ,LTACQ  ,CASDSF,LCASDS                         FILES
      INTEGER   SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT        FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF                     FILES
C                                                                               FILES
C         INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                           IXCASF
      COMMON   /IXCASF/                                                         IXCASF
     1  IXCASF(1),LIXCAS
C                                                                               IXCASF
C         INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)              IXCDSF
      COMMON   /IXCDSF/                                                         IXCDSF
     1    IXCDSF(1),LIXCDS
C                                                                               IXCDSF
C         LEM CONTROL CARD INPUT DATA                                           LEMCM
      COMMON   /LEMCM /                                                         LEMCM
     1    TITLE(10)      ,ICASE  ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2   ,ENDR   ,ENDZ   ,ISTG   ,ICAMS  ,IYES   ,IACQ   ,ICLASS,ISEXT  ,ISCC   LEMCM
     3   ,ICAS2  ,ICAS3  ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4   ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
```

```
     5   ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                       LEMCM
      DIMENSION  RSEED(7)                                                  LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5         LEMCM
     1   ,RSEED6,RSEED7                                                    LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                        LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR   ,ENDZ                           LEMCM
C                                                                          LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                             PAGECM
      COMMON  /PAGECM/                                                     PAGECM
     1    NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                           PAGECM
C                                                                          PAGECM
C        STATISTICAL INFORMATION FOR LEM                                   STATS
      COMMON  /STATS /                                                     STATS
     1    ITER   ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR               STATS
      EQUIVALENCE  ( NT,ITER )                                             STATS
C                                                                          STATS
C       DEBUGGING PRINT FLAG                                               CAS3
      COMMON  /DEBUGF/  DEBUGF                                             CAS3
C                                                                          CAS3
C                                                                          CAS3
C     LOCAL VARIABLES                                                      CAS3
C        I       DO LOOP INDEX                                             CAS3
C        IREG    REGION INDEX (1,2,...,NREG)                               CAS3
C        IZONE   ZONE INDEX (1,2,...,NZONES)                               CAS3
C                                                                          CAS3
C                                                                          CAS3
      IF ( M1K2KC .EQ. 0.0 )  GO TO 990                                    JULY76
C                                                                          CAS3
C        INITIALIZE REGION, ZONE, AND STRATA POINTERS FOR CAS              CAS3
C        INTERMEDIATE FILE. ( REGION RECORDS ARE 3-12, ZONE RECORDS ARE    CAS3
C        13-62, STRATA RECORDS ARE 63-387 )                                CAS3
 110  IRREG= 2                                                             CAS3
      IRZONE= 12                                                           CAS3
      IRSTR = 62                                                           CAS3
      IREG= 0                                                              CAS3
C                                                                          CAS3
C        INITIALIZE DATA SET 13 (COUNTRY LEVEL)                            CAS3
      DO 130 I=1,LDS13                                                     CAS3
      DSET13(I)= 0.0                                                       CAS3
 130  CONTINUE                                                             CAS3
C                                                                          CAS3
```

```
C       INITIALIZE DATA SET 12 (REGION LEVEL)                              CAS3
 140  IREG= IREG + 1                                                         CAS3
      DO 150 I=1,LDS12                                                       CAS3
      DSET12(I)= 0.0                                                         CAS3
 150  CONTINUE                                                               CAS3
      IZONE= 0                                                               CAS3
      ENDREG= 0                                                              CAS3
C        SET NLINE TO CAUSE PAGE EJECT BEFORE PRINTING NEXT REGION           CAS3
C        ON AREA AND PRODUCTION SUMMARY REPORT                               CAS3
      NLINE= MXLINE + 1                                                      CAS3
C                                                                            CAS3
C        READ DATA SET 8 (REGION LEVEL) FROM CAS INTERMEDIATE FILE           CAS3
      IRREG= IRREG + 1                                                       CAS3
      CALL RANACF (CASDSF,IRREG,DSET8,LCASDS,IXCDSF,LIXCDS,1)                CAS3
C                                                                            JULY76
      HWAR12= HWAR2                                                          JULY76
C                                                                            JULY76
      IF ( M1M2ZR .NE. 0 ) GO TO 180                                         JULY76
C        M1M2ZR = 0. NO ZONE IN REGION HAS AT LEAST 2 ACQUIRED SEGMENTS      JULY76
C        USE ESTIMATED GROUP 1,2 VARIANCE AND HISTORICAL GROUP 1,2           JULY76
C        WHEAT AREA FROM COUNTRY LEVEL.                                      JULY76
      ESTVR = ESTVC                                                          JULY76
      ANALVR = ANALVC                                                        JULY76
      HWAR12 = HWAC2                                                         JULY76
C                                                                            CAS3
C        INITIALIZE DATA SET 11 (ZONE LEVEL)                                 CAS3
 180  IZONE = IZONE + 1                                                      CAS3
      IF ( IZONE .EQ. NZONES )  ENDREG= 1                                    CAS3
      DO 190 I=1,LDS11                                                       CAS3
      DSET11(I)= 0.0                                                         CAS3
 190  CONTINUE                                                               CAS3
C                                                                            CAS3
C        READ DATA SET 7 (ZONE LEVEL) FROM CAS INTERMEDIATE FILE             CAS3
      IRZONE= IRZONE + 1                                                     CAS3
      CALL RANACF (CASDSF,IRZONE,DSET7,LCASDS,IXCDSF,LIXCDS,1)               CAS3
C                                                                            JULY76
      HWAZ12= HWAZ2                                                          JULY76
      IF ( M1K2KZ .GT. 1 )  GO TO 200                                        JULY76
C                                                                            JULY76
C        LESS THAN 2 ACQUIRED SEGMENTS IN ZONE. USE ESTIMATED GROUP 1,2      JULY76
C        VARIANCE AND HISTORICAL GROUP 1,2 WHEAT AREA FROM REGION OR         JULY76
```

C-6

```
C         COUNTRY LEVEL.                                                     JULY76
      ESTVZ = ESTVR                                                          JULY76
      ANALVZ = ANALVR                                                        JULY76
      HWAZ12= HWAR12                                                         JULY76
      IF ( HWAR12 .EQ. 0.0 )  GO TO 200                                      JULY76
      WRATIO = ( HWAZ3/HWAR12 )**2                                           JULY76
      AVARZ = ESTVR*WRATIO                                                   JULY76
      ANAVZ = ANALVR*WRATIO                                                  JULY76
C                                                                            CAS3
C         COMPUTE DATA SET 10 FOR EACH STRATA IN THIS ZONE AND AGGREGATE     CAS3
C         INTO DATA SET 11 AT THE ZONE LEVEL.                                CAS3
  200 CALL DS10                                                              JULY76
      IF ( NFATAL .NE. 0 )  GO TO 990                                        CAS3
C                                                                            CAS3
C         GENERATE REST OF DATA SET 11 (ZONE LEVEL)                          CAS3
C         EQS. 109, 113, 115 - 117                                           CAS3
      AERRZ = EWAZ - TWAZ                                                    CAS3
      PRERRZ= EPRODZ - TPRODZ                                                CAS3
      IF ( TWAZ .NE. 0.0 )  TYZ= TPRODZ/TWAZ                                 CAS3
      IF ( EWAZ .NE. 0.0 )  EYZ= EPRODZ/EWAZ                                 CAS3
      IF ( TYZ .NE. 0.0 )  YERRZ= ( EYZ - TYZ )/TYZ *100.0                   CAS3
C                                                                            CAS3
C                                                                            CAS3
C         TEMPORARY DEBUGGING PRINTOUT                                       CAS3
C                                                                            CAS3
C                                                                            CAS3
C         AGGREGATE ZONE DATA SET 11 UP TO DATA SET 12 (REGION LEVEL)        CAS3
C         EQS. 125 - 127, 129 - 131, 133, 137 - 143                          CAS3
      DO 440 I=1,7                                                           CAS3
      DSET12(I)= DSET12(I) + DSET11(I)                                       CAS3
  440 DSET12(I+12)= DSET12(I+12) + DSET11(I+12)                              CAS3
      PRVARR= PRVARR + PRVARZ                                                CAS3
C                                                                            CAS3
C         ON FIRST ITERATION AND FIRST PREDICTION POINT, SKIP READING        CAS3
C         CAS CUMULATIVE FILE.                                               CAS3
      IF ( NT .EQ. 1 .AND. IPP .EQ. 1 )  GO TO 450                           CAS3
C                                                                            CAS3
C         READ DATA SET 15 (ZONE DATA) FROM CAS CUMULATIVE FILE              CAS3
C         NOTE ...  EQUIVALENCE  ( DSET15,CASCUM(5) )                        CAS3
      CALL RWCASF (IRZONE,CASCUM,1)                                          CAS3
C                                                                            CAS3
```

```
C         ACCUMULATE ZONE DATA IN DATA SET 15 (CAS CUMULATIVE FILE)       CAS3
      IF ( NT .GT. 1 )  GO TO 470                                          CAS3
C         FIRST ITERATION.  CLEAR DATA SET 15 BEFORE ACCUMULATING          CAS3
  450 ICASC(1)= REGION                                                     CAS3
      ICASC(2)= ZONE                                                       CAS3
      CASCUM(3)= 0.0                                                       CAS3
      ICASC(4)= NSTRAZ                                                     CAS3
      DO 460 I=1,LDS15                                                     CAS3
      DSET15(I)= 0.0                                                       CAS3
  460 CONTINUE                                                             CAS3
  470 DO 480 I=1,19                                                        CAS3
      DSET15(I)= DSET15(I) + DSET11(I)                                     CAS3
  480 CONTINUE                                                             CAS3
C         EQS. 173 - 175                                                   CAS3
      SQAERZ= SQAERZ + AERRZ**2                                            CAS3
      SQPERZ= SQPERZ + PRERRZ**2                                           CAS3
      SQYERZ= SQYERZ + YERRZ**2                                            CAS3
C                                                                          CAS3
C                                                                          CAS3
C         TEMPORARY DEBUGGING PRINTOUT                                     CAS3
C                                                                          CAS3
C                                                                          CAS3
C         WRITE DATA SET 15 (ZONE DATA) BACK ONTO  CAS CUMULATIVE FILE     CAS3
      CALL RWCASF (IRZONE,CASCUM,2)                                        CAS3
C                                                                          CAS3
C         UPDATE ZONE DATA ON CAS DISTRIBUTION FILE                        CAS3
      IF ( DISTFF .NE. 0 )  CALL RWDISF (2,DSET11)                         CAS3
C                                                                          CAS3
C         PRINT AREA AND PRODUCTION SUMMARY REPORT DATA FOR THIS ZONE      CAS3
      IF ( PRINTF .NE. 0 .AND. APREP .NE. 0 )  CALL CASOUT (-2)            CAS3
C         TEST FOR END OF REGION                                           CAS3
      IF ( IZONE .LT. NZONES )  GO TO 180                                  CAS3
C                                                                          CAS3
C         GENERATE REST OF DATA SET 12 (REGION LEVEL)                      CAS3
C         EQS. 128, 132, 134 - 136                                         CAS3
      AERRR= EWAR - TWAR                                                   CAS3
      PRERRR= EPRODR - TPRODR                                              CAS3
      IF ( TWAR .NE. 0.0 )  TYR= TPRODR/TWAR                               CAS3
      IF ( EWAR .NE. 0.0 )  EYR= EPRODR/EWAR                               CAS3
      IF ( TYR .NE. 0.0 )  YERRR= ( EYR - TYR )/TYR *100.0                 CAS3
C                                                                          CAS3
```

```
C                                                                        CAS3
C        TEMPORARY DEBUGGING PRINTOUT                                    CAS3
C                                                                        CAS3
C        AGGREGATE REGION DATA SET 12 UP TO DATA SET 13 (COUNTRY LEVEL)  CAS3
C        EQS. 144 - 146, 148 - 150, 152, 156 - 162                       CAS3
      DO 540 I=1,7                                                       CAS3
      DSET13(I)= DSET13(I) + DSET12(I)                                   CAS3
  540 DSET13(I+12)= DSET13(I+12) + DSET12(I+12)                          CAS3
      PRVARC= PRVARC + PRVARR                                            CAS3
C                                                                        CAS3
C        ON FIRST ITERATION AND FIRST PREDICTION POINT, SKIP READING     CAS3
C        CAS CUMULATIVE FILE.                                            CAS3
      IF ( NT .EQ. 1 .AND. IPP .EQ. 1 )  GO TO 550                       CAS3
C                                                                        CAS3
C        READ DATA SET 16 (REGION DATA) FROM CAS CUMULATIVE FILE         CAS3
C        NOTE ...  EQUIVALENCE  ( DSET16,CASCUM(5) )                     CAS3
      CALL RWCASF (IRREG,CASCUM,1)                                       CAS3
C                                                                        CAS3
C        ACCUMULATE REGION DATA IN DATA SET 16 (CAS CUMULATIVE FILE)     CAS3
      IF ( NT .GT. 1 )  GO TO 570                                        CAS3
C        FIRST ITERATION. CLEAR DATA SET 16 BEFORE ACCUMULATING          CAS3
  550 ICASC(1)= REGION                                                   CAS3
      ICASC(2)= 0                                                        CAS3
      ICASC(3)= 0                                                        CAS3
      ICASC(4)= 0                                                        CAS3
      DO 560 I=1,LDS16                                                   CAS3
      DSET16(I)= 0.0                                                     CAS3
  560 CONTINUE                                                           CAS3
C                                                                        CAS3
  570 DO 580 I=1,19                                                      CAS3
      DSET16(I)= DSET16(I) + DSET12(I)                                   CAS3
  580 CONTINUE                                                           CAS3
C        EQS. 176 - 178                                                  CAS3
      SQAERR= SQAERR + AERRR**2                                          CAS3
      SQPERR= SQPERR + PRERRR**2                                         CAS3
      SQYERR= SQYERR + YERRR**2                                          CAS3
C                                                                        CAS3
C                                                                        CAS3
C        TEMPORARY DEBUGGING PRINTOUT                                    CAS3
C                                                                        CAS3
C                                                                        CAS3
```

```
C        WRITE DATA SET 16 (REGION DATA) BACK ONTO CAS CUMULATIVE FILE  CAS3
      CALL RWCASF (IRREG,CASCUM,2)                                      CAS3
C                                                                       CAS3
C        UPDATE REGION DATA ON CAS DISTRIBUTION FILE                    CAS3
      IF ( DISTFF .NE. 0 )  CALL RWDISF (1,DSET12)                      CAS3
C                                                                       CAS3
C        PRINT AREA AND PRODUCTION SUMMARY REPORT DATA FOR THIS REGION  CAS3
      IF ( PRINTF .NE. 0 .AND. APREP .NE. 0 )  CALL CASOUT (-1)         CAS3
C        TEST FOR END OF COUNTRY                                        CAS3
      IF ( IREG .LT. NREGS )  GO TO 140                                 CAS3
C                                                                       CAS3
C        GENERATE REST OF DATA SET 13 (COUNTRY LEVEL)                   CAS3
C        EQS. 147, 151, 153 - 155                                       CAS3
      AERRC= EWAC - TWAC                                                CAS3
      PRERRC= EPRODC - TPRODC                                           CAS3
      IF ( TWAC .NE. 0.0 )  TYC= TPRODC/TWAC                            CAS3
      IF ( EWAC .NE. 0.0 )  EYC= EPRODC/EWAC                            CAS3
      IF ( TYC .NE. 0.0 )  YERRC= ( EYC - TYC )/TYC *100.0              CAS3
C                                                                       CAS3
C        COMPUTE CONFIDENCE LEVELS                                      CAS3
      CALL CONFL                                                        CAS3
C                                                                       CAS3
C                                                                       CAS3
C        TEMPORARY DEBUGGING PRINTOUT                                   CAS3
C                                                                       CAS3
C                                                                       CAS3
C        ON FIRST ITERATION AND FIRST PREDICTION POINT, SKIP READING    CAS3
C        CAS CUMULATIVE FILE.                                           CAS3
      IF ( NT .EQ. 1 .AND. IPP .EQ. 1 )  GO TO 650                      CAS3
C                                                                       CAS3
C        READ DATA SET 17 (COUNTRY DATA) FROM CAS CUMULATIVE FILE       CAS3
C        NOTE ...  EQUIVALENCE ( DSET17,CASCUM(5) )                     CAS3
      CALL RWCASF (2,DSET17,1)                                          CAS3
C                                                                       CAS3
C        ACCUMULATE COUNTRY DATA IN DATA SET 17 (CAS CUMULATIVE FILE)   CAS3
      IF ( NT .GT. 1 )  GO TO 670                                       CAS3
C        FIRST ITERATION. CLEAR DATA SET 17 BEFORE ACCUMULATING         CAS3
  650 DO 660 I=1,LDS17                                                  CAS3
      DSET17(I)= 0.0                                                    CAS3
  660 CONTINUE                                                          CAS3
C                                                                       CAS3
```

```
  670 DO 680 I=1,19                                                      CAS3
      DSET17(I)= DSET17(I) + DSET13(I)                                   CAS3
  680 CONTINUE                                                           CAS3
C        EQS. 179 - 181                                                  CAS3
      SQAERC= SQAERC + AERRC**2                                          CAS3
      SQPERC= SQPERC + PRERRC**2                                         CAS3
      SQYERC= SQYERC + YERRC**2                                          CAS3
C        ACCUMULATE CONFIDENCE LEVELS ALSO.                              CAS3
      DO 690 I=20,25                                                     CAS3
      DSET17(I+3)= DSET17(I+3) + DSET13(I)                               CAS3
  690 CONTINUE                                                           CAS3
C                                                                        CAS3
C                                                                        CAS3
C        TEMPORARY DEBUGGING PRINTOUT                                    CAS3
C                                                                        CAS3
C                                                                        CAS3
C                                                                        CAS3
C        WRITE DATA SET 17 (COUNTRY DATA) BACK ONTO CAS CUMULATIVE FILE CAS3
      CALL RWCASF (2,DSET17,2)                                           CAS3
C                                                                        CAS3
C        UPDATE COUNTRY DATA ON CAS DISTRIBUTION FILE                    CAS3
      IF ( DISTFF .NE. 0 )  CALL RWDISF (0,DSET13)                       CAS3
C                                                                        CAS3
C      COMPUTE MEAN VALUES AND PRINT AREA AND PRODUCTION SUMMARY REPORTCAS3
      IF ( PRINTF .NE. 0 )  CALL CASOUT (0)                              CAS3
C                                                                        CAS3
C        CLOSE CAS INTERMEDIATE FILE                                     CAS3
C                                                                        CAS3
C        ON THE FINAL ITERATION (UNLESS NTRIAL = 1), COMPUTE             CAS3
C        CLWA AND CLPRD IN DATA SET 18.                                  CAS3
      IF ( NT .EQ. NTRIAL .AND. NT .GT. 1 )  CALL DS18                   CAS3
C                                                                        CAS3
  990 RETURN                                                             CAS3
C                                                                        CAS3
C                                                                        CAS3
      END                                                                CAS3
```

```
      SUBROUTINE CLASS(SEED2,TYPE,WINDOW,M,BCC,SIGCC,XI)                CLASS
C                                                                       CLASS
C  THIS SUBROUTINE CALCULATES THE INPUT CLASSIFICATION ERROR FOR        CLASS
C     TRAINING SEGMENTS, AND THE TOTAL CLASSIFICATION ERROR.            CLASS
C                                                                       CLASS
      COMMON/CAMERR/ COUN2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,             CAMERR
     1  PW(3,4),BERR(3,4),SIGERR(3,4)                                   CAMERR
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,            TRAINS
     1   TIWIN(4,25),TITOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),         TRAINS
     1   TPTRUE,TIZULU(4),TPEST(4),TPERR(4),TERTOT(3),TM(3),TV(3),TB(3) TRAINS
      INTEGER TIZULU                                                    TRAINS
      DIMENSION ITRAIN(129)                                             TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                         TRAINS
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CAMS CONTROL CARD INPUT DATA                                   CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,        CAMSCM
     1    IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                    CAMSCM
      REAL MS                                                           CAMSCM
C                                                                       CAMSCM
      COMMON/ERROR/TITL(4),IDATE,PESTIM,TOT,ALOCAL,ERTOT(3)             ERROR
     1   ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,       ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                         ERROR
      DIMENSION IERS(40)                                                ERROR
      EQUIVALENCE(TITL,IERS)                                            ERROR
      REAL MULT                                                         ERROR
      INTEGER TID,CROPD                                                 ERROR
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
      INTEGER TYPE, WINDOW                                              CLASS
      REAL M                                                            CLASS
      DOUBLE PRECISION SEED2                                            CLASS
C                                                                       CLASS
C  ADD INPUT CLASSIFICATION ERROR TO CROP CALENDAR ERROR                CLASS
      B=BERR(TYPE,WINDOW)+BCC                                           CLASS
      SIG=SQRT(SIGERR(TYPE,WINDOW)*SIGERR(TYPE,WINDOW)+SIGCC*SIGCC)     CLASS
C                                                                       CLASS
C  COMPUTE XBAR AND SIGMA, THEN XI                                      CLASS
      XBAR=PW(TYPE,WINDOW)*(1. +M *B)                                   CLASS
      SIGMA=PW(TYPE,WINDOW) *M*SIG                                      CLASS
      CALL FETAD (SEED2,XBAR,SIGMA,XI,0,IER)                            CLASS
      NARG=1                                                            CLASS
      IARG(1)=IER                                                       CLASS
      IF(IER.GT.0) CALL ERRMES(4HCAMS,5HCLASS,4,0)                      CLASS
      IF(IER.GE.3)  XI=XBAR                                             CLASS
C                                                                       CLASS
C  ERROR REPORT                                                         CLASS
   10 CONTINUE                                                          CLASS
      IF(PRINTF.LE.0.OR.IREP.LE.0)RETURN                                CLASS
      TV(TYPE)=0.                                                       CLASS
```

```
      IF(SIGMA.EQ.0.) GO TO 12                                          CLASS
      TV(TYPE)=(XI-XBAR)/(PW(TYPE,WINDOW)*M)                            CLASS
   12 CONTINUE                                                          CLASS
      CLBIAS(TYPE)=B                                                    CLASS
      CLRAND(TYPE)=TV(TYPE)                                             CLASS
      CLTOT(TYPE)=B+TV(TYPE)                                            CLASS
      ERTOT(TYPE)=M*(B+TV(TYPE))                                        CLASS
      ERBIAS(TYPE)=M*B                                                  CLASS
      ERRAND(TYPE)=M*TV(TYPE)                                           CLASS
      RETURN                                                            CLASS
      END                                                               CLASS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FOR,IS CLASSN
      SUBROUTINE CLASSN
C     THIS ROUTINE CONTROLS THE COMPUTATION OF CLASS NUMBERS AND THE    *******
C     GENERATION OF ISUBH2 FILE FROM THE SUBHST FILE.                   *******
      COMMON/FILES1/                                                    FILES1
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                          CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1    H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2   ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS       CASFLG
     3   ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
     4   ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5   ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR        CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C         CONSTANT QUANTITIES FOR LEM PROGRAM                           CONST
      COMMON  /CONST /                                                  CONST
     1    NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                      CONST
C                                                                       CONST
C         LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1    TITLE(10)    ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2   ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3   ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4   ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5   ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                     LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1   ,RSEED6,RSEED7                                                 LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C         INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE  IXSUBH
      COMMON  /IXSUBH/                                                  IXSUBH
     1 LIXSSH,IXSUBH(1)                                                 MOD1
C                                                                       IXSUBH
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ         FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO        FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD        FILES
```

```
     4  ,INP    ,OUTP   ,TACQ   ,LTACQ ,CASDSF,LCASDS                    FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF               FILES
C                                                                        FILES
C          CAS DATA SETS 1,2, AND 3                                      DSET1
      COMMON  /DSET1 /                                                   DSET1
     1   ISUBST,TWAK   ,HWAK   ,EWAK   ,M1K    ,CT1K   ,ANALVK,EPWK   ,EPW2K  JULY76
     2   ,SMPKPI,SUMPK2,SUMPK  ,KSUB   ,NCLASS                           JULY76
      REAL   M1K , M2K                                                   JULY76
      DIMENSION  DSET1(14), DSET2(14), DSET3(6)                          JULY76
      EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                          DSET1
     1   , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                               DSET1
C                                                                        DSET1
C          ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
C          TABLES NECCESSARY TO DETERMINE CLASS SETS WITHIN A ZONE       CLSTAB
      COMMON /CLSTAB/                                                    CLSTAB
     1       ISTRAT(300),ISBSTR(300),NSCNT(300),IGROUP(300),IDAT1(300),  MOD1
     2       IDAT2(300),XORD(300),IXPT(300),IRANK(300),IBPT(10),IEPT(10),MOD1
     3.      MAXCLS,ICLCNT,ISUB1,NACQ                                    CLSTAB
      DIMENSION DAT1(300),DAT2(300),RANK(300)                            MOD1
      EQUIVALENCE (IDAT1(1),DAT1(1)),(IDAT2(1),DAT2(1)),(IRANK(1),       CLSTAB
     1RANK(1))                                                           CLSTAB
C          SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                    SSHDTA
      COMMON  /SSHDTA/                                                   SSHDTA
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW  SSHDTA
     2  ,AREAK ,PWK    ,NAGR   ,NA     ,DELTPW,DELTPM,CV1   ,CV2   ,CV3  SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                               JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                     JULY76
      DIMENSION  SSHDTA(39)                                              JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                     SSHDTA
C                                                                        SSHDTA
      DIMENSION IBUF(39)                                                 *******
      DIMENSION BUF(39)                                                  MOD1
      EQUIVALENCE (BUF(1),IBUF(1))                                       MOD1
      DIMENSION IDUM(150)
      DATA MAXSCT/300/,IRCT/19/                                          MOD1
```

```
      ISUB = 1                                                          *******
      NORD = 0                                                          *******
      DO 5  I=1,39                                                      *******
      IBUF(I) = 0                                                       *******
      SSHDTA(I) = 0.0                                                   *******
    5 CONTINUE                                                          *******
      IFIRST = 1                                                        *******
      IF(IPP .NE. 1)GO TO 15                                            *******
      MAXCLS = 9                                                        *******
      CALL RANACF(ISUBH2,0,0,0,IXSUBH,LIXSSH,0)                         *******
   15 ISUB1 = 0                                                         *******
      NACQ = 0                                                          *******
      DO 20  I=1,MAXSCT                                                 *******
      ISTRAT(I) = 0                                                     *******
      ISBSTR(I) = 0                                                     *******
      NSCNT(I) = 0                                                      *******
      IGROUP(I) = 0                                                     *******
      IDAT1(I) = 0                                                      *******
      IDAT2(I) = 0                                                      *******
   20 CONTINUE                                                          *******
   22 IF(NORD .EQ. 0)GO TO 25                                           *******
      NORD = 0                                                          *******
      IF(IPP .EQ. 1)GO TO 30                                            *******
      GO TO 45                                                          *******
   25 IF(IPP .NE. 1)GO TO 40                                            *******
      READ(SUBHST)(IBUF(I),I=1,6),(IDUM(J),J=1,IMXSEG),(IBUF(K),K=8,19)
      BUF(19) = 1.2
      IF(IFIRST .EQ. 0)GO TO 27                                         *******
      IFIRST = 0                                                        *******
      GO TO 30                                                          *******
   27 IF(IBUF(2) .NE. IREG2 .OR.  IBUF(3) .NE. IZONE2)GO TO 80          *******
   30 DO 35  I=1,IRCT                                                   *******
      SSHDTA(I) = BUF(I)                                                MOD1
   35 CONTINUE                                                          *******
      GO TO 54                                                          MOD1
   40 CALL RANACF(ISUBH2,ISUB,IBUF   ,LSUBH2,IXSUBH,LIXSSH,1)           *******
      IF(IFIRST .EQ. 0)GO TO 42                                         *******
      IFIRST = 0                                                        *******
      GO TO 45                                                          *******
   42 IF(IBUF(2) .NE. IREG2 .OR. IBUF(3) .NE. IZONE2)GO TO 80           *******
   45 DO 50  I=1,IRCT                                                   *******
```

```
      SSHDTA(I) = BUF(I)                                                MOD1
   50 CONTINUE                                                          *******
   54 M1K = 0                                                           MOD1
      IF(NSEG .NE. 0)CALL GROUP                                         MOD1
      MXK = M1K                                                         MOD1
   55 ISUB = ISUB + 1                                                   *******
      ISUB1 = ISUB1 + 1                                                 *******
      IF(ISUB1 .GT. MAXSCT)GO TO 200                                    *******
      ISTRAT(ISUB1) = ISTRA2                                            *******
      ISBSTR(ISUB1) = ISUBS2                                            *******
      NSCNT(ISUB1) = MXK                                                *******
      IGROUP(ISUB1) = GRPNO                                             *******
      DAT1(ISUB1) = AREAK                                               *******
      DAT2(ISUB1) = HISTPW/100.0                                        *******
      ISM1 = ISUB - 1                                                   *******
      IF(IPP .EQ. 1)CALL RANACF(ISUBH2,ISM1,SSHDTA,LSUBH2,IXSUBH,LIXSSH,*******
     12)                                                                *******
      GO TO 22                                                          *******
   80 NURD = 1                                                          *******
      NACQ = 0                                                          *******
      DO 85 I=1,ISUB1                                                   *******
      NACQ = NACQ + NSCNT(I)                                            *******
   85 CONTINUE                                                          *******
      IF(NACQ .GT. MAXSCT)GO TO 200                                     *******
      IF(NACQ .LT. 2)GO TO 90                                           *******
      IF(NACQ .LT. 2*H)GO TO 95                                         *******
      GO TO 100                                                         *******
   90 CALL ASSCLS(1)                                                    *******
      GO TO 105                                                         *******
   95 CALL ASSCLS(2)                                                    *******
      GO TO 105                                                         *******
  100 CALL SEGTAB                                                       *******
      CALL DETCLS                                                       *******
      CALL ASSCLS(0)                                                    *******
  105 I     = ISUB - ISUB1                                              *******
      DO 110 J=1,ISUB1                                                  *******
      CALL RANACF(ISUBH2,I,SSHDTA,LSUBH2,IXSUBH,LIXSSH,1)               *******
      CLASS(IPP) = IDAT2(J)                                             *******
      CALL RANACF(ISUBH2,I,SSHDTA,LSUBH2,IXSUBH,LIXSSH,2)               *******
      I = I + 1                                                         *******
  110 CONTINUE                                                          *******
```

```
      IF(BUF(1) .EQ. ENDFIL)GO TO 150                                  MOD1
      IF(ENDR .EQ. 0)GO TO 15                                          *******
      IF(IBUF(2) - ENDR)15,115,145                                     *******
  115 IF(ENDZ .EQ. 0)GO TO 15                                          *******
      IF(IBUF(3) .LE. ENDZ)GO TO 15                                    *******
  145 BUF(1) = ENDFIL                                                  MOD1
  150 IF(IPP .NE. 1)GO TO 152                                          MOD1
      CALL RANACF(ISUBH2,ISUB,IBUF,LSUBH2,IXSUBH,LIXSSH,2)             *******
  152 REWIND CAMSF                                                     MOD1
      DO 155 I=1,NCAMSK                                                *******
      READ(CAMSF)                                                      *******
  155 CONTINUE                                                         *******
      NRCAMS = NCAMSK - 1                                              *******
      GO TO 210                                                        *******
  200 NFATAL = NFATAL + 1                                              *******
      WRITE(OUTP,900)IREG2,IZONE2                                      *******
  210 IF(NFATAL .EQ. 0)GO TO 250                                       *******
      WRITE(OUTP,901)                                                  *******
      STOP                                                             *******
  250 RETURN                                                           *******
  900 FORMAT(1H0,48HEITHER TOO MANY SUBSTRATA OR SEGMENTS IN REGION-,I4,MOD1
     16H ZONE-,I4)                                                     MOD1
  901 FORMAT(1H ,43HFATAL ERRORS IN PASS 0 OF CAS. RUN ABORTED.)       *******
      END                                                              *******
```

```
      SUBROUTINE CONFL                                                  CONFL
C        COMPUTES CONFIDENCE LEVELS IN DATA SET 13                      CONFL
C                                                                       CONFL
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        CAS DATA SET 13 (COUNTRY DATA -- SECOND PASS)                  DSET13
      COMMON  /DSET13/                                                  DSET13
     1   HWAC  ,TWAC  ,EWAC  ,AERRC ,AVARC ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2  ,TYC   ,EYC   ,YERRC ,M1C   ,M2C   ,CT1C  ,CT2C  ,CT3C  ,ANAVC  DSET13
     3  ,ANPRVC,CLEWA ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC               DSET13
      REAL  M1C ,M2C                                                    DSET13
      DIMENSION  DSET13(25)                                             DSET13
      EQUIVALENCE  ( DSET13,HWAC )                                      DSET13
C                                                                       DSET13
C     LOCAL VARIABLES                                                   CONFL
C        X      = ARGUMENT FOR P(X) FUNCTION                            CONFL
C        X1     = ARGUMENT FOR P(X) FUNCTION                            CONFL
C        X2     = ARGUMENT FOR P(X) FUNCTION                            CONFL
C        Y      = VARIANCE QUANTITY USED TO CALCULATE X, X1, OR X2      CONFL
C                                                                       CONFL
C                                                                       CONFL
C        COMPUTE CLEWA (EQ. 163)                                        CONFL
      ARG(1)= 5HAVARC                                                   CONFL
      IARG(2)= 163                                                      CONFL
      Y= YSUB(AVARC,EWAC)                                               CONFL
      X= 0.1*EWAC/Y                                                     CONFL
      CLEWA= ( 2.0*PSUB(X) - 1.0 )*100.0                                CONFL
C                                                                       CONFL
C        COMPUTE CLATEC (EQ. 166)                                       CONFL
      X1= ( EWAC - 0.9*TWAC )/Y                                         CONFL
      X2= ( EWAC - 1.1*TWAC )/Y                                         CONFL
      CLATEC= ( PSUB(X1) - PSUB(X2) )*100.0                             CONFL
C                                                                       CONFL
C        COMPUTE CLEPRD (EQ. 165)                                       CONFL
      ARG(1)= 6HEPRODC                                                  CONFL
      IARG(2)= 165                                                      CONFL
      Y= YSUB(PRVARC,EPRODC)                                            CONFL
      X= 0.1*EPRODC/Y                                                   CONFL
      CLEPRD= ( 2.0*PSUB(X) - 1.0 )*100.0                               CONFL
C                                                                       CONFL
C        COMPUTE CLPTEC (EQ. 167)                                       CONFL
      X1= ( EPRODC - 0.9*TPRODC )/Y                                     CONFL
      X2= ( EPRODC - 1.1*TPRODC )/Y                                     CONFL
      CLPTEC= ( PSUB(X1) - PSUB(X2) )*100.0                             CONFL
C                                                                       CONFL
C        COMPUTE CLATWC  (EQ. 168)                                      CONFL
      ARG(1)= 5HANAVC                                                   CONFL
      IARG(2)= 168                                                      CONFL
      Y= YSUB(ANAVC,TWAC)                                               CONFL
      X1= ( EWAC - 0.9*TWAC )/Y                                         CONFL
      X2= ( EWAC - 1.1*TWAC )/Y                                         CONFL
      CLATWC= ( PSUB(X1) - PSUB(X2) )*100.0                             CONFL
C                                                                       CONFL
C        COMPUTE CLPTWC  (EQ. 169)                                      CONFL
```

```
      ARG(1)= 6HANPRVC                                                  CONFL
      IARG(2)= 169                                                      CONFL
      Y= YSUB(ANPRVC,TPRODC)                                            CONFL
      X1= ( FPRODC - 0.9*TPRODC )/Y                                     CONFL
      X2= ( FPRODC - 1.1*TPRODC )/Y                                     CONFL
      CIPTWF= ( PSUB(X1) - PSUB(X2) )*100.0                             CONFL
C                                                                       CONFL
  900 RETURN                                                            CONFL
      END                                                               CONFL
```

```
@FLT CORREL,1,760427, 39048   , 1

      SUBROUTINE CORREL(ITMAX,ACQNO,WINDOW,IUSE)                         CORREL
C                                                                        CORREL
C  THIS SUBROUTINE TRIES TO CORRELATE A TRAINING SEGMENT WITH THE        CORREL
C     ORDINARY SEGMENT BEING PROCESSED, USING THE PRIORITY LIST.         CORREL
C                                                                        CORREL
      COMMON/SEGTRU/COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,               SEGTRU
     1   IT,IPRIOR(6),ISPW,PT(2)                                         SEGTRU
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,             TRAINS
     1   ITWIN(4,25),ITTOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),          TRAINS
     1   TPTRUE,ITZULU(4),IPEST(4),TPERR(4),TERTOT(3),TM(3),TV(3),TB(3)  TRAINS
      INTEGER ITZULU                                                     TRAINS
      DIMENSION ITRAIN(129)                                              TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                          TRAINS
C        FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                   FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO          FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                       FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT   FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                 FILES
C                                                                        FILES
      COMMON/TACQ/IHOLD(4,129),IHP                                       TACQ
      INTEGER ACQNO,WINDOW                                               CORREL
      IUSE=1                                                             CORREL
   60 CONTINUE                                                           CORREL
      IF(IPRIOR(IUSE).NE.0.AND.IUSE.LE.6) GO TO 10                       CORREL
   65 CONTINUE                                                           CORREL
      IUSE=7                                                             CORREL
      RETURN                                                             CORREL
   10 CONTINUE                                                           CORREL
      CALL TSAVE(IPRIOR(IUSE),1,IBAD)                                    CORREL
      IF(IBAD.GT.0) GO TO 65                                             CORREL
C                                                                        CORREL
C  FIND FIRST ACQUIS.DATE ON TRAINING SEGMENT .LE. ACQUIS                CORREL
      ITACQ=0                                                            CORREL
      DO 20 I=1,WINDOW                                                   CORREL
      DO 30 J=1,25                                                       CORREL
      IF(ITWIN(I,J).EQ.0) GO TO 20                                       CORREL
      IF(ACQNO.LT.ITWIN(I,J)) GO TO 40                                   CORREL
      ITACQ=ITWIN(I,J)                                                   CORREL
      ISAVE=I                                                            CORREL
      JSAVE=J                                                            CORREL
   30 CONTINUE                                                           CORREL
   20 CONTINUE                                                           CORREL
      IF(ITWIN(WINDOW,25).NE.0) ITACQ=ITWIN(WINDOW,25)                   CORREL
   40 CONTINUE                                                           CORREL
      IF(ITACQ.EQ.0) GO TO 50                                            CORREL
      DO 43 I=1,3                                                        CORREL
      TM(I)=TMM(I,ISAVE,JSAVE)                                           CORREL
      TB(I)=TBB(I,ISAVE,JSAVE)                                           CORREL
      TV(I)=TVV(I,ISAVE,JSAVE)                                           CORREL
   43 CONTINUE                                                           CORREL
C                                                                        CORREL
C  CHECK IF OK TRAINING ACQUIS.                                          CORREL
      IDIFF=ACQNO-ITACQ                                                  CORREL
      IF(IDIFF.GT.ITMAX) GO TO 50                                        CORREL
      DO 45 I=1,3                                                        CORREL
```

ORIGINAL PAGE IS
OF POOR QUALITY

28234-6029-RU-00
Page 427

```
      TERTOT(I)=TM(I)*(TB(I)+TV(I))                                     CORREL
   45 CONTINUE                                                          CORREL
      RETURN                                                            CORREL
   50 CONTINUE                                                          CORREL
      IUSE=IUSE+1                                                       CORREL
      GO TO 60                                                          CORREL
      END                                                               CORREL
```

@ FLT CROP,1,760427, 39052   , 1

```
      SUBROUTINE CROP(SEED4,TYPE,SEASON,WINDOW,IFIRST,BCC,SIGCC          CROP
     1   ,ITSEG)                                                         CROP
C                                                                        CROP
C   THIS SUBROUTINE CALCULATES THE CROP CALENDAR ERROR FOR TRAINING      CROP
C       SEGMENTS.                                                        CROP
C                                                                        CROP
C       CAMS CONTROL CARD INPUT DATA                                     CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGFX,ISKIP,ITMAX,IPEP,IWIND,        CAMSCM
     1     IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                    CAMSCM
      REAL MS                                                            CAMSCM
C                                                                        CAMSCM
      COMMON/CROPW/COUN3,IREG3,IZONE3,ISTRA3,ISUB3,                      CROPW
     1  START(2,4),END(2,4),SD(2),ERR(2,5)                               CROPW
      INTEGER START,END,SD,ERR                                           CROPW
      COMMON/ERROR/TITL(4),IDATE,PESTIM,IOT,ALOCAL,ERTOT(3)              ERROR
     1    ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,       ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                          ERROR
      DIMENSION IERS(40)                                                 ERROR
      EQUIVALENCE(TITL,IERS)                                             ERROR
      REAL MULT                                                          ERROR
      INTEGER TID,CROPD                                                  ERROR
      DOUBLE PRECISION SEED4                                             CROP
      INTEGER TYPE,SEASON,WINDOW                                         CROP
C                                                                        CROP
C  COMPUTE DELTA T                                                       CROP
      IF(IFIRST.GT.1.OR.TYPE.GT.1) GO TO 10                              CROP
      CALL BETAD(SEED4,0.,0.,RN,1,IER)                                   CROP
      ITSEG=IFIX(RN)*SD(SEASON)                                          CROP
C                                                                        CROP
C  COMPUTE CROP CALENDAR ERROR                                           CROP
   10 CONTINUE                                                           CROP
      CROPD=ITSEG -ERR(SEASON,WINDOW)                                    CROP
      DELTA= FLOAT(CROPD) /FLOAT(END(SEASON,WINDOW)-                     CROP
     1    START(SEASON,WINDOW) + 1)                                      CROP
      BCC=G(TYPE,SEASON,1)*DELTA+G(TYPE,SEASON,2)*DELTA*DELTA            CROP
      SIGCC=ABS(H(TYPE,SEASON,1)*DELTA+ H(TYPE,SEASON,2)*DELTA*DELTA)    CROP
      RETURN                                                             CROP
      END                                                                CROP
```

```
FOR,IS DETCLS
      SUBROUTINE DETCLS                                                 DETCLS
C     THIS ROUTINE DETERMINS HOW MANY CLASSES THE X ARRAY XORD CONTAINS DETCLS
C     AND ASSIGNS THE CLASS BOUNDARY POINTS WITHIN XORD.                DETCLS
C         TABLES NECCESSARY TO DETERMINE CLASS SETS WITHIN A ZONE       CLSTAB
      COMMON /CLSTAB/                                                   CLSTAB
     1        ISTRAT(300),ISBSTR(300),NSCNT(300),IGROUP(300),IDAT1(300), MOD1
     2       IDAT2(300),XORD(300),IXPT(300),IRANK(300),IBPT(10),IEPT(10),MOD1
     3        MAXCLS,ICLCNT,ISUB1,NACQ                                  CLSTAB
      DIMENSION DAT1(300),DAT2(300),RANK(300)                           MOD1
      EQUIVALENCE (IDAT1(1),DAT1(1)),(IDAT2(1),DAT2(1)),(IRANK(1),      CLSTAB
     1RANK(1))                                                          CLSTAB
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                          CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
      DIMENSION ID(10),IDUM(10)                                         DETCLS
C                                                                       DETCLS
C    ******                                                             DETCLS
      DATA CC/.25/                                                      MOD1
C     THIS DATA IS A BUILT IN CONSTANT, IT CAN BE MODIFIED VIA COMPILATIDETCLS
C    ******                                                             DETCLS
C                                                                       DETCLS
      IH = H                                                            DETCLS
      DO 2 M=1,10                                                       MOD1
      IBPT(M) = 0                                                       MOD1
      IEPT(M) = 0                                                       MOD1
    2 CONTINUE                                                          MOD1
      CO = CC/NACQ                                                      MOD1
      ICLCNT = 0                                                        DETCLS
      K = 1                                                             DETCLS
    5 I = 1                                                             DETCLS
    7 IF((NACQ -K) .NE. IRANK(I))GO TO 25                               DETCLS
      LB = 0                                                            DETCLS
      IUB = NACQ                                                        DETCLS
      IF(ICLCNT .LE. 0)GO TO 20                                         DETCLS
```

```
      DO 15 J=1,ICLCNT                                       DETCLS
      IF(ID(J) .LT. I)GO TO 10                               DETCLS
      IUB = MINO(IUB,ID(J))                                  DETCLS
      GO TO 15                                               DETCLS
   10 LB = MAXO(LB,ID(J))                                    DETCLS
   15 CONTINUE                                               DETCLS
   20 IP = IXPT(I)                                           DETCLS
      IP1 = IXPT(I+1)                                        DETCLS
      IUBP = IXPT(IUB)                                       DETCLS
      LBP1 = IXPT(LB + 1)                                    DETCLS
      IF((I - LB) .LT. IH)GO TO 25                           DETCLS
      IF((IUB - I) .LT. IH)GO TO 25                          DETCLS
      IF((XORD(IP)- XORD(LBP1)) .LT. CO)GO TO 25             MOD1
      IF((XORD(IUBP) - XORD(IP1)) .LT. CO)GO TO 25           MOD1
      ICLCNT = ICLCNT + 1                                    DETCLS
      ID(ICLCNT) = I                                         DETCLS
      IF(ICLCNT .GT. MAXCLS)GO TO 30                         DETCLS
   25 I = I + 1                                              DETCLS
      IF(I .LE. (NACQ-1))GO TO 7                             DETCLS
      K = K + 1                                              DETCLS
      IF(K .LE. (NACQ-1))GO TO 5                             DETCLS
   30 IF(ICLCNT .EQ. 0)GO TO 40                              DETCLS
      DO 35 I=1,ICLCNT                                       DETCLS
      IDUM(I) = I                                            DETCLS
   35 CONTINUE                                               DETCLS
      CALL SORTAG(ID,1,ICLCNT,IDUM)                          DETCLS
   40 IBPT(1) = 1                                            DETCLS
      IMAX = ICLCNT + 1                                      DETCLS
      IEPT(IMAX) = NACQ                                      DETCLS
      IF(ICLCNT .EQ. 0)GO TO 55                              DETCLS
      I = 2                                                  DETCLS
      DO 50 J=1,ICLCNT                                       DETCLS
      IBPT(I) = ID(I-1) + 1                                  DETCLS
      IEPT(J) = ID(I-1)                                      DETCLS
      I = I + 1                                              DETCLS
   50 CONTINUE                                               DETCLS
   55 ICLCNT = ICLCNT + 1                                    DETCLS
   60 RETURN                                                 DETCLS
      END                                                    DETCLS
```

```
 FOR,IS DS10
      SUBROUTINE DS10                                                    DS10
C        PROCESSES DATA SET 10 AT THE STRATA LEVEL                       DS10
C                                                                        DS10
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                       CASCM
      COMMON /CASCM /                                                    CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)   ,PPRUTS(5,2)   ,YPRUTS(3,2) CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)       CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)            CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE         CASCM
C                                                                        CASCM
C        DATA BLOCK FOR CAS CUMULATIVE FILE                              CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                                CASCUM
      COMMON  /CASCUM/                                                   CASCUM
     1   CASCUM(32),   BUFFR(504)                                        CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)           CASCUM
     1  ,DSET17(28)                                                      CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                      CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )             CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                     CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                     CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                     CASCUM
C                                                                        CASCUM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                   CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                        CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                   CASFLG
C                                                                        CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                              CNTRL
      COMMON /CNTRL /                                                    CNTRL
     1   PRINTF,NSTART,SEED(7)                                           CNTRL
      INTEGER  PRINTF                                                    CNTRL
      DOUBLE PRECISION  SEED                                             CNTRL
C                                                                        CNTRL
C        CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                     DSET4
      COMMON  /DSET4 /                                                   DSET4
     1   STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                 JULY76
```

```
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                        JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS    ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS   ,ANVARS                 JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                       JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                              JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )       DSET4
C                                                                           DSET4
C        CAS DATA SET 7 (AT ZONE LEVEL)                                     DSET7
      COMMON  /DSET7 /                                                      DSET7
     1   ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3      JULY76
     2  ,ESTVZ ,HWAZ12                                                      JULY76
     3  ,M1K2CL(10)   ,EPWCL(10)    ,EPW2CL(10)   ,PKPICL(10)               JULY76
     4  ,PK2CL(10)    ,PKCL(10)     ,SSQ(10)                                JULY76
      INTEGER  ZONE                                                         JULY76
      REAL  M1K2KZ, M1K2CL                                                  JULY76
      DIMENSION  DSET7(81)                                                  JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                           DSET7
C                                                                           DSET7
C        CAS DATA SET 8 (AT REGION LEVEL)                                   DSET8
      COMMON  /DSET8 /                                                      DSET8
     1   REGION,HWAR2 ,ER    ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR      JULY76
     2  ,M1M2ZR,FILL8(71)                                                   JULY76
      INTEGER  REGION                                                       JULY76
      REAL  M1K2KR                                                          JULY76
      DIMENSION  DSET8(10)                                                  JULY76
      EQUIVALENCE  ( DSET8,REGION )                                         DSET8
C                                                                           DSET8
C        CAS DATA SET 9 (AT COUNTRY LEVEL)                                  DSET9
      COMMON  /DSET9 /                                                      DSET9
     1   COUNTR,HWAC2 ,EC    ,M1K2KC,ANALVC,M1M2ZC,HWAC1 ,EWAC1 ,ESTVC      JULY76
      INTEGER  COUNTR                                                       JULY76
      REAL  M1K2KC                                                          JULY76
      DIMENSION  DSET9(9)                                                   JULY76
      EQUIVALENCE  ( DSET9,COUNTR )                                         DSET9
C                                                                           DSET9
C        CAS DATA SET 10 (STRATA DATA -- FINAL PASS)                        JULY76
      COMMON  /DSET10/                                                      JULY76
     1   HWAS  ,TWAS  ,EWAS  ,AERRS ,AVARS ,TPRODS,EPRODS,PRERRS,PRVARS     JULY76
     2  ,YS    ,ESTYS ,YERRS ,M1JS  ,M2JS  ,CT1S  ,CT2S  ,CT3S  ,ANAVS      JULY76
     3  ,ANPRVS,ES                                                          JULY76
```

```
      REAL  M1JS , M2JS                                                 JULY76
      DIMENSION  DSET10(20)                                             JULY76
      EQUIVALENCE  ( DSET10,HWAS )                                      JULY76
C                                                                       DSET10
C        CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                      JULY76
      COMMON  /DSET11/                                                  DSET11
     1   HWAZ  ,TWAZ   ,EWAZ   ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ    ,EYZ    ,YERRZ ,M1Z   ,M2Z    ,CT1Z   ,CT2Z   ,CT3Z   ,ANAVZ  DSET11
     3  ,ANPRVZ                                                         DSET11
      REAL  M1Z , M2Z                                                   DSET11
      DIMENSION  DSET11(19)                                             DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                      DSET11
C                                                                       DSET11
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                    IXCASF
      COMMON  /IXCASF/                                                  IXCASF
     1  IXCASF(1),LIXCAS
C                                                                       IXCASF
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)       IXCDSF
      COMMON  /IXCDSF/                                                  IXCDSF
     1   IXCDSF(1),LIXCDS
C                                                                       IXCDSF
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C       DEBUGGING PRINT FLAG                                            DS10
      COMMON  /DEBUGF/  DEBUGF                                          DS10
C                                                                       DS10
C                                                                       DS10
C     LOCAL VARIABLES                                                   DS10
C        D      = INTERMEDIATE QUANTITY USED TO COMPUTE TAU2S           DS10
```

```
C                  (D = 0 OR 1 DEPENDING UPON NO. OF ACQUIRED GROUP 1,2  DS10
C                   SEGMENTS IN STRATA)                                   DS10
C       DENOM   = DENOMINATOR OF RATIO USED TO COMPUTE TAU2S AND SIGM2S  DS10
C       F       = INTERMEDIATE QUANTITY USED TO COMPUTE ANAVS            DS10
C                 (= ANALV AT STRATA, ZONE, REGION, OR COUNTRY LEVEL)    DS10
C       FM      = NHISTY (IN FLOATING POINT)                             DS10
C       HWAS12 = HWAS1 + HWAS2                                           DS10
C       HWA12   = HWA1J + HWA2J (WHERE J = S, Z, R, OR C)                DS10
C       I       = DO LOOP INDEX                                          DS10
C       ISTRAZ = ZONE INDEX (1,2,....,NSTRAZ)                            DS10
C       MYV12   = MYV1J + MYV2J  (WHERE J = S, Z, R, OR C)               DS10
C       NU      = DO LOOP (YEAR) INDEX IN MULTI-YEAR VARIANCE LOOP       DS10
C       RATIO   = INTERMEDIATE QUANTITY USED TO COMPUTE TAU2S AND SIGM2SDS10
C       RN1     = RANDOM NUMBER IN NUMERATORS OF RATIOS IN EXPRESSIONS  DS10
C                 FOR TAU2S AND SIGM2S                                   DS10
C       RN2     = RANDOM NUMBER IN DENOMINATORS OF RATIOS IN EXPRESSIONSDS10
C                 FOR TAU2S AND SIGM2S                                   DS10
C       SRMV12 = SQRT. OF MULTI-YEAR VARIANCE FOR GROUP 1,2 SEGMENTS    DS10
C                 AT STRATA, ZONE, REGION, OR COUNTRY LEVELS             DS10
C       SRMYV3 = SQRT. OF MULTI-YEAR VARIANCE FOR GROUP 3 SEGMENTS      DS10
C                 AT STRATA, ZONE, REGION, OR COUNTRY LEVELS             DS10
C       SUM1    = SUM OF TERMS IN EXPRESSION FOR TAU2S BEFORE DIVIDING  DS10
C                 BY M.                                                  DS10
C       SUM2    = FIRST SUM IN EXPRESSION FOR SIGM2S (SUM OF RATIO**2)  DS10
C       SUM3    = SECOND SUM IN EXPRESSION FOR SIGM2S (SUM OF RATIOS)   DS10
C       TERM1   = INTERMEDIATE QUANTITY USED TO COMPUTE PRVARS AND      DS10
C                 ANPRVS (=EVYRS + ESTYS**2)                             DS10
C       TERM2   = INTERMEDIATE QUANTITY USED TO COMPUTE PRVARS          DS10
C                 (=EVYRS*EWAS**2)                                       DS10
C                                                                        DS10
      REAL MYV12                                                         DS10
C                                                                        DS10
C                                                                        DS10
C        COMPUTE INTERMEDIATE QUANTITY USED TO COMPUTE AVARZ LATER.ON.  JULY76
      CONZ = 1.0                                                         JULY76
      IF ( HWAZ12 .NE. 0.0 )  CONZ= 1.0 + HWAZ3/HWAZ12                   JULY76
C                                                                        JULY76
      ISTRAZ= 0                                                          DS10
C        INITIALIZE DATA SET 10 (STRATA LEVEL)                           DS10
  200 ISTRAZ= ISTRAZ + 1                                                 DS10
      DO 210 I=1,LDS10                                                   DS10
```

```
      DSET10(I)= 0.0                                                    DS10
  210 CONTINUE                                                          DS10
C                                                                       DS10
C        READ DATA SETS 4,5, AND 6 FROM CAS INTERMEDIATE FILE           DS10
      IRSTR = IRSTR + 1                                                 DS10
      CALL RANACF (CASDSF,IRSTR,DSET4,LCASDS,IXCDSF,LIXCDS,1)           DS10
C                                                                       DS10
C        GENERATE DATA SET 10 (STRATA LEVEL)                            DS10
C                                                                       DS10
C        MOVE YS, ESTYS, M1JS, M2JS, CT1S, CT2S, AND CT3S               DS10
C        FROM DATA SETS 4,5,6 TO DATA SET 10                            DS10
      YS    = XYS                                                       DS10
      ESTYS= XESTYS                                                     DS10
      M1JS = XM1JS                                                      DS10
      M2JS = XM2JS                                                      DS10
      CT1S = XCT1S                                                      DS10
      CT2S = XCT2S                                                      DS10
      CT3S = XCT3S                                                      DS10
      HWAS12= HWAS1 + HWAS2                                             DS10
      HWAS = HWAS12 + HWAS3                                             JULY76
C                                                                       DS10
C        COMPUTE ES  (EQ. 90)                                           DS10
      IF( M1JS+M2JS .EQ. 0.0 .OR. HWAS12 .EQ. 0.0 )  GO TO 232          DS10
C        M1JS + M2JS .GE. 1  (AT LEAST ONE ACQUIRED SEGMENT IN STRATA)  DS10
      ES= ( EWAS1 + EWAS2 ) / HWAS12                                    DS10
      GO TO 240                                                         DS10
C        M1JS + M2JS = 0.  NO ACQUIRED SEGMENTS IN STRATA.              DS10
  232 IF ( M1K2KZ .EQ. 0.0 )  GO TO 234                                 DS10
C        SUM OF M1K + M2K .GE. 1.  AT LEAST ONE ACQUIRED SEGMENT IN ZONEDS10
      ES= EZ                                                            DS10
      GO TO 240                                                         DS10
C        M1K2KZ = 0.  NO ACQUIRED SEGMENTS IN ZONE                      DS10
  234 IF ( M1K2KR .EQ. 0.0 )  GO TO 236                                 DS10
C        SUM OF M1K + M2K .GE. 1.  AT LEAST ONE ACQUIRED SEGMENT IN REG.DS10
      ES= ER                                                            DS10
      GO TO 240                                                         DS10
C        M1K2KR = 0.  NO ACQUIRED SEGMENTS IN REGION                    DS10
  236 ES= EC                                                            DS10
C                                                                       DS10
  240 IF ( M1M2ZC .EQ. 0.)  GO TO 320                                   DS10
C                                                                       DS10
```

```
      IF ( M1K2KZ .LT. 2.0 )  GO TO 250                                JULY76
      IF ( M1JS + M2JS .GT. 0.0 )  GO TO 260                           JULY76
C        COMPUTE AREA VARIANCE AND ANALYTIC AREA VARIANCE OF STRATA     JULY76
C        WITHOUT ANY ACQUIRED SEGMENTS ( OR STRATA IN A ZONE WITH LESS  JULY76
C        THAN 2 ACQUIRED SEGMENTS)                                      JULY76
C        (THE AREA VARIANCE OF STRATA WITH SEGMENTS HAS ALREADY BEEN    JULY76
C        COMPUTED IN SUBROUTINE CAS2)                                   JULY76
  250 IF ( HWAZ12 .EQ. 0.0 )  GO TO 320                                JULY76
      WRATIO = ( HWAS/HWAZ12 )**2                                      JULY76
      AVARS = WRATIO*ESTVZ                                             JULY76
      ANAVS = WRATIO*ANALVZ                                            JULY76
      GO TO 320                                                        JULY76
C                                                                      JULY76
C        AT LEAST ONE ACQUIRED SEGMENT IN STRATA AND AT LEAST TWO       JULY76
C        IN ZONE.                                                       JULY76
C        ADD CONTRIBUTION OF THIS STRATA TO ZONE AREA VARIANCE          JULY76
  260 AVARS = VARS                                                     JULY76
      ANAVS = ANVARS                                                   JULY76
      VZMULT= ( CONZ + HWAS3/HWAS12 )**2                               JULY76
      AVARZ = AVARZ + V1V2S*VZMULT                                     JULY76
      ANAVZ = ANAVZ + (ANVS1+ANVS2)*VZMULT                             JULY76
C                                                                      DS10
C        COMPUTE HWAS, TWAS, ... , ANPRVS  (EQS. 94-105)                DS10
  320 HWAS= HWAS12 + HWAS3                                             DS10
      TWAS= TWAS1 + TWAS2 + TWAS3                                      DS10
      EWAS= EWAS1 + EWAS2 + ES*HWAS3                                   DS10
      AERRS= EWAS - TWAS                                               DS10
      TPRODS= YS*TWAS                                                  DS10
      EPRODS= ESTYS*EWAS                                               DS10
      PRERRS= EPRODS - TPRODS                                          DS10
      TERM1 = ESTYS*ESTYS - EVYRS                                      JULY76
      TERM2 = EWAS*EWAS*EVYRS                                          DS10
      PRVARS= AVARS*TERM1 + TERM2                                      DS10
      IF ( YS .GT. 0.0 )  YERRS= ( ESTYS - YS )/YS *100.0              DS10
      ANPRVS= ANAVS*TERM1 + TERM2                                      DS10
C                                                                      DS10
C                                                                      DS10
C        TEMPORARY DEBUGGING PRINTOUT                                   DS10
C                                                                      DS10
C                                                                      DS10
C        AGGREGATE STRATA DATA SET 10 UP TO DATA SET 11 (ZONE LEVEL)    DS10
```

```
C          EQS. 106 - 108, 110 - 112, 114, 118 - 124                     DS10
      DO 340 I=1,4                                                     JULY76
      DSET11(I)= DSET11(I) + DSET10(I)                                 JULY76
      DSET11(I+5)= DSET11(I+5) + DSET10(I+5)                           JULY76
  340 DSET11(I+12)= DSET11(I+12) + DSET10(I+12)                        JULY76
      CT3Z = CT3Z + CT3S                                               JULY76
      ANPRVZ= ANPRVZ + ANPRVS                                          JULY76
C                                                                        DS10
C          ON FIRST ITERATION AND FIRST PREDICTION POINT, SKIP READING   DS10
C          CAS CUMULATIVE FILE.                                          DS10
      IF ( NT .EQ. 1 .AND. IPP .EQ. 1 )  GO TO 350                       DS10
C                                                                        DS10
C          READ DATA SET 15 (STRATA DATA) FROM CAS CUMULATIVE FILE       DS10
C          NOTE ...  EQUIVALENCE  ( DSET14,CASCUM(5) )                   DS10
      CALL RWCASF (IRSTR,CASCUM,1)                                       DS10
C                                                                        DS10
C          ACCUMULATE STRATA DATA IN DATA SET 14 (CAS CUMULATIVE FILE)   DS10
      IF ( NT .GT. 1 )  GO TO 370                                        DS10
C          FIRST ITERATION.  CLEAR DATA SET 14 BEFORE ACCUMULATING       DS10
  350 ICASC(1)= REGION                                                   DS10
      ICASC(2)= ZONE                                                     DS10
      ICASC(3)= STRATA                                                   DS10
      ICASC(4)= 0                                                        DS10
      DO 360 I=1,LDS14                                                   DS10
      DSET14(I)= 0.0                                                     DS10
  360 CONTINUE                                                           DS10
C                                                                        DS10
  370 DO 380 I=1,19                                                      DS10
      DSET14(I)= DSET14(I) + DSET10(I)                                 JULY76
  380 CONTINUE                                                           DS10
C          EQS. 170 - 172                                                DS10
      SQAERS= SQAERS + AERRS**2                                          DS10
      SQPERS= SQPERS + PRERRS**2                                         DS10
      SQYERS= SQYERS + YERRS**2                                          DS10
C                                                                        DS10
C                                                                        DS10
C          TEMPORARY DEBUGGING PRINTOUT                                  DS10
C                                                                        DS10
C                                                                        DS10
C          WRITE DATA SET 14 BACK ONTO CAS CUMULATIVE FILE               DS10
      CALL RWCASF (IRSTR,CASCUM,2)                                       DS10
```

```
C                                                                        DS10
      IF ( PRINTF .NE. 0 .AND. APREP .NE. 0 )  CALL CASOUT (ISTRAZ)      DS10
C        TEST FOR END OF ZONE                                            DS10
      IF ( ISTRAZ .LT. NSTRAZ )  GO TO 200                               DS10
C                                                                        DS10
 990  RETURN                                                             DS10
      END                                                                DS10
```

```
FOR,IS DS123
      SUBROUTINE DS123                                                  DS123
C        PROCESSES DATA SETS 1, 2, AND 3 AT THE SUBSTRATA LEVEL         DS123
C                                                                       DS123
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                      CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)    CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1   , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                   CASCUM
     2   , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                   CASCUM
     3   , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                   CASCUM
C                                                                       CASCUM
C        DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)             CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFR )                                     CASDSB
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1   , ( PIK,CASDSB(121) )                                          CASDSB
C                                                                       CASDSB
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW    ,IBW    ,WINDOW,IPD    ,IPP    ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS        CASFLG
     3  ,ENDC   ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
```

```
     4  ,LDS1   ,LDS4   ,LDS7   ,LDS8   ,LDS9   ,LDS10 ,LDS11 ,LDS12 ,LDS13   CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR               CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                        CASFLG
C                                                                             CASFLG
C        CONTROL PARAMETERS FOR LEM PROGRAM                                   CNTRL
      COMMON  /CNTRL /                                                        CNTRL
     1    PRINTF,NSTART,SEED(7)                                               CNTRL
      INTEGER  PRINTF                                                         CNTRL
      DOUBLE PRECISION  SEED                                                  CNTRL
C                                                                             CNTRL
C        CAS DATA SETS 1,2, AND 3                                             DSET1
      COMMON  /DSET1 /                                                        DSET1
     1    ISUBST,TWAK   ,HWAK   ,EWAK   ,M1K    ,CT1K   ,ANALVK,EPWK   ,EPW2K  JULY76
     2  ,SMPKPI,SUMPK2,SUMPK ,KSUB   ,NCLASS                                  JULY76
      REAL  M1K , M2K                                                         JULY76
      DIMENSION  DSET1(14), DSET2(14), DSET3(6)                               JULY76
      EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                               DSET1
     1  , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                                     DSET1
C                                                                             DSET1
C        CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                          DSET4
      COMMON  /DSET4 /                                                        DSET4
     1    STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                     JULY76
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                          JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS    ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS   ,ANVARS                    JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                         JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                                JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )         DSET4
C                                                                             DSET4
C        CAS DATA SET 7 (AT ZONE LEVEL)                                       DSET7
      COMMON  /DSET7 /                                                        DSET7
     1    ZONE   ,HWAZ2 ,EZ     ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3     JULY76
     2  ,ESTVZ ,HWAZ12                                                        JULY76
     3  ,M1K2CL(10)   ,EPWCL(10)    ,EPW2CL(10)   ,PKPICL(10)                 JULY76
     4  ,PK2CL(10)    ,PKCL(10)     ,SSQ(10)                                  JULY76
      INTEGER  ZONE                                                           JULY76
      REAL  M1K2KZ, M1K2CL                                                    JULY76
      DIMENSION  DSET7(81)                                                    JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                             DSET7
C                                                                             DSET7
```

```
C        FILE DEFINITIONS AND RECORD LENGTHS                               FILES
       COMMON  /FILES /                                                    FILES
      1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
      2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO          FILES
      3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
      4   ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                       FILES
       INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT    FILES
      1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                  FILES
C                                                                          FILES
C        INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE      IXSUBH
       COMMON  /IXSUBH/                                                    IXSUBH
      1 LIXSSH,IXSUBH(1)                                                   MOD1
C                                                                          IXSUBH
       COMMON/FILES1/                                                      FILES1
      1ISUBH2,LSUBH2,MXCLSS                                                FILES1
C        SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                        SSHDTA
       COMMON  /SSHDTA/                                                    SSHDTA
      1    COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW  SSHDTA
      2   ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3     SSHDTA
      3   ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                               JULY76
       INTEGER  GRPNO , CLASS , RDSSH                                      JULY76
       DIMENSION  SSHDTA(39)                                               JULY76
       EQUIVALENCE  ( SSHDTA, COUN2 )                                      SSHDTA
C                                                                          SSHDTA
C       DEBUGGING PRINT FLAG                                               DS123
       COMMON  /DEBUGF/  DEBUGF                                            DS123
C                                                                          DS123
C                                                                          DS123
       EQUIVALENCE ( IER,IARG(1) )                                         DS123
C                                                                          DS123
C      LOCAL VARIABLES                                                     DS123
C         CONK    = QUANTITY WHICH IS CONSTANT FOR A GIVEN SUBSTRATA       DS123
C                   INDEPENDENT OF SEGMENT (=NK*RK*AREAPS WHERE NK = NAGR)DS123
C         FNK     = NK = NAGR (FL. PT.)                                    DS123
C         I       = DO LOOP INDEX                                          DS123
C         LEVEL   = ERROR LEVEL (=0 FOR NON-FATAL ERROR, =1 FOR FATAL)     DS123
C         M2      = M2K (INTEGER)                                          DS123
C         PWKNEY = SUBSTRATA PROPORTION WHEAT (PW) FOR NON-EPOCH YEAR.     DS123
C         RK      = RK  (EQ. 1)                                            DS123
C         RKSQ    = (RK*AREAPS)**2                                         DS123
C         SIGMA   = STANDARD DEVIATION FOR BETA DISTRIBUTION ROUTINE.      DS123
```

```
C                                                                         DS123
C        ARRAY FOR SAVING HISTPW, PWK, AND AREAK BEFORE CONVERTING        JULY76
C        FROM PERCENT TO FRACTION.                                        JULY76
      DIMENSION  SSHSAV(3)                                                JULY76
C                                                                         JULY76
C                                                                         DS123
      DO 210 I=2,LDS1                                                     DS123
      DSET1(I)= 0.0                                                       DS123
 210  CONTINUE                                                            DS123
C                                                                         DS123
      ISUBST= ISUBS2                                                      DS123
      SSHSAV(1) = HISTPW                                                  JULY76
      SSHSAV(2) = PWK                                                     JULY76
      SSHSAV(3) = AREAK                                                   JULY76
C        CONVERT  HISTPW AND PW TO FRACTIONS FROM PERCENT.                DS123
      HISTPW= 0.01*HISTPW                                                 DS123
      PWK= 0.01*PWK                                                       DS123
C        CONVERT SUBSTRATA LAND AREA TO HECTARES FROM KM**2               DS123
      AREAK=  100.0*AREAK                                                 DS123
C        EQ. 1                                                            DS123
      RK= AREAK/( FLOAT(NA)*AREAPS )                                      DS123
      RKSQ= (RK*AREAPS)**2                                                DS123
      FNK= NAGR                                                           DS123
      CONK= FNK*RK*AREAPS                                                 DS123
C        EQS. 7, 8, AND 10                                                DS123
      TWAK= CONK*PWK                                                      DS123
      HWAK= CONK*HISTPW                                                   DS123
      CT1K= 1.0                                                           DS123
C                                                                         DS123
C        TEST GROUP NUMBER AND GENERATE DATA SET 1, 2, OR 3               DS123
      IF ( GRPNO - 2 )  310,350,390                                       DS123
C                                                                         DS123
C        GROUP I SUBSTRATUM                                               DS123
C        CHECK THE NUMBER OF SEGMENTS IN THIS SUBSTRATUM                  DS123
 310  IF ( NSEG .GT. 0 )  GO TO 320                                       DS123
C        FATAL ERROR.  NO SEGMENTS IN SUBSTRATUM                          DS123
      CALL ERRMES (3HCAS,5HDS123,13,1)                                    DS123
      GO TO 990                                                           DS123
C                                                                         DS123
C        PROCESS ALL SEGMENTS IN THIS GROUP I SUBSTRATUM.                 DS123
C        EQS. 2A - 6A                                                     DS123
```

```
 320  CALL GROUP                                                         DS123
      IF ( NFATAL .NE. 0 )  GO TO 990                                    DS123
C        WERE ANY GROUP I SEGMENTS ACQUIRED FOR THIS SUBSTRATUM          DS123
      IF ( M1K .EQ. 0.0 )  GO TO 390                                     DS123
C        GENERATE REST OF DATA SET 1  (EQS. 9, 11, AND 12)               DS123
      EWAK= CONK*EPWK/M1K                                                DS123
      VMULTK= ( FNK - M1K )*RKSQ*FNK/M1K                                 DS123
      ANALVK= VMULTK*(PWK*CV2)**2                                        DS123
C                                                                        DS123
C        AGGREGATE SUBSTRATA DATA SET 1 INTO DATA SET 4 (STRATA LEVEL)   DS123
C        ( TWAK,HWAK,EWAK,M1K,MYVK,VMULTK,CT1K,ANALVK )                  DS123
C        (EQS. 19-25, 37)                                                DS123
      DO 345 I=2,7                                                       JULY76
      DSET4(I)= DSET4(I) + DSET1(I)                                      DS123
 345  CONTINUE                                                           DS123
      GO TO 385                                                          JULY76
C                                                                        DS123
C        GROUP II SUBSTRATUM                                             DS123
C        CHECK THE NUMBER OF SEGMENTS IN THIS SUBSTRATUM                 DS123
 350  IF ( NSEG .EQ. 0 )  GO TO 360                                      DS123
C        PROCESS ALL GROUP II SEGMENTS IN THIS SUBSTRATA                 DS123
C        EQS. 2B - 6B                                                    DS123
      CALL GROUP                                                         DS123
      IF ( NFATAL .NE. 0 )  GO TO 990                                    DS123
C                                                                        DS123
C        GENERATE REST OF DATA SET 2                                     DS123
 360  IF ( TOPT .EQ. 0 )  GO TO 370                                      DS123
C                                                                        JULY76
C        COMPUTE NON-EPOCH YEAR WHEAT AREA                               JULY76
      KSUB = KSUB + 1                                                    JULY76
      WAKNEY(KSUB) = CV4*HWAK                                            JULY76
      HWA2K(KSUB) = HWAK                                                 DS123
C                                                                        DS123
C        EQN. 16  ( DEFER DIVISION BY HWAK UNTIL AFTER ANALVK COMPUTED )DS123
 370  VMULTK= ( FNK*FNK - FNK )*RKSQ                                     DS123
C        EQN. 17                                                         DS123
      ANALVK= VMULTK*( PWK*CV2 )**2                                      DS123
      VMULTK= VMULTK/HWAK                                                DS123
C                                                                        DS123
C        AGGREGATE SUBSTRATA DATA SET 2 INTO DATA SET 5 (STRATA LEVEL)   DS123
C        EQNS. 31,32,34,35,36,38,40                                      DS123
```

```
      DO 380 I=1,6                                                      JULY76
      DSET5(I   )= DSET5(I   ) + DSET2(I+1)                             DS123
  380 CONTINUE                                                          DS123
C        NOTE... AT THIS POINT EWAS2 HAS NOT BEEN COMPUTED YET AND      DS123
C        VMULTK AND ANVS2 (EQS. 36 AND 40) ARE INCOMPLETE               JULY76
C        (COMPLETED IN SUBROUTINE CAS2)                                 JULY76
      P2IDPK= P2IDPK + EPWK/HISTPW                                      DS123
C                                                                       JULY76
  385 M1K2CL(NCLASS)= M1K2CL(NCLASS) + M2K                              JULY76
      EPWCL(NCLASS) = EPWCL(NCLASS)  + EPWK                             JULY76
      EPW2CL(NCLASS)= EPW2CL(NCLASS) + EPW2K                            JULY76
      PKPICL(NCLASS)= PKPICL(NCLASS) + SMPKPI                           JULY76
      PK2CL(NCLASS) = PK2CL(NCLASS)  + SUMPK2                           JULY76
      PKCL(NCLASS)  = PKCL(NCLASS)   + SUMPK                            JULY76
C        RESTORE ORIGINAL VALUES BEFORE WRITING BACK ONTO ISUBH2 FILE   JULY76
      HISTPW = SSHSAV(1)                                                JULY76
      PWK    = SSHSAV(2)                                                JULY76
      AREAK  = SSHSAV(3)                                                JULY76
C        WRITE SUBSTRATA DATA BACK ONTO ISUBH2 FILE                     JULY76
      CALL RANACF (ISUBH2,NRSSH,SSHDTA,LSUBH2,IXSUBH,LIXSSH,2)          JULY76
      GO TO 990                                                         DS123
C                                                                       DS123
C        GROUP III SUBSTRATUM.  SET GROUP III FLAG                      DS123
C        AGGREGATE SUBSTRATA DATA SET 3 INTO DATA SET 6 (STRATA LEVEL)  DS123
C        EQS. 47 - 49                                                   DS123
  390 TWAS3= TWAS3 + TWAK                                               DS123
      HWAS3= HWAS3 + HWAK                                               DS123
      XCT3S = XCT3S  + CT3K                                             DS123
C                                                                       DS123
C                                                                       DS123
C                                                                       DS123
C            TEMPORARY DEBUGGING PRINTOUT                               DS123
  990 CONTINUE
C                                                                       DS123
C                                                                       DS123
      RETURN                                                            DS123
      END                                                               DS123
```

@ ELT DS18,1,760427, 39088 , 1

```
      SUBROUTINE DS18                                                   DS18
C        COMPUTES CLWA AND CLPRD IN DATA SET 18 ON THE FINAL ITERATION. DS18
C                                                                       DS18
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C        SUMMARY DATA FOR REPORTS                                       SUMDTA
      COMMON  /SUMDTA/                                                  SUMDTA
     1   CVAEPT,CVEPTA,SDPER ,CVPEPT,CVEPTP,CSUMR(18,18)                SUMDTA
C                                                                       SUMDTA
C                                                                       DS18
C     INPUT DATA ...                                                    DS18
C                                                                       DS18
C        NT = ITER = MONTE CARLO ITERATION NUMBER                       DS18
C        IPP       = PREDICTION POINT INDEX (INCLUDING BIOWINDOWS)      DS18
C        SQAERC    = SUM OF SQUARES OF AREA ERRORS                      DS18
C        SQPERC    = SUM OF SQUARES OF PRODUCTION ERRORS                DS18
C        DSET17(2) = TRUE WA                                            DS18
C        DSET17(3) = MEAN ESTIMATED WA                                  DS18
C        DSET17(4) = MEAN AREA ERROR                                    DS18
C        DSET17(6) = TRUE PRODUCTION                                    DS18
C        DSET17(7) = MEAN ESTIMATED PRODUCTION                          DS18
C        DSET17(8) = MEAN PRODUCTION ERROR                              DS18
C                                                                       DS18
C     OUTPUT QUANTITIES ...                                             DS18
C        CSUMR(11,IPP) = CLWA                                           DS18
C        CSUMR(15,IPP) = CLPRD                                          DS18
C                                                                       DS18
C     LOCAL VARIABLES ...                                               DS18
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C          FNT   = NT = MONTE CARLO ITERATION (FLOATING POINT)           DS18
      C          FNT1  = NT - 1                                                DS18
      C          VEAC  = AREA VARIANCE ERROR FOR COUNTRY                       DS18
      C          VEPC = PRODUCTION VARIANCE ERROR FOR COUNTRY                  DS18
      C          Y     = DIVISOR FOR X1 AND X2                                 DS18
      C          X1    = ARGUMENT FOR P(X) FUNCTION                            DS18
      C          X2    = ARGUMENT FOR P(X) FUNCTION                            DS18
      C                                                                        DS18
      C                                                                        DS18
      C          EQ. 182                                                       DS18
            FNT= NT                                                            DS18
            FNT1= NT - 1                                                       DS18
            VEAC= ( SOAERC - (DSET17(4)**2)*FNT )/ FNT1                        DS18
            ARG(1)= 4HVEAC                                                     DS18
            IARG(2)= 182                                                       DS18
            Y= YSUB(VEAC,DSET17(2))                                            DS18
            X1= ( DSET17(3) - 0.9*DSET17(2) )/ Y                               DS18
            X2= ( DSET17(3) - 1.1*DSET17(2) )/ Y                               DS18
      C          STORE CLWA IN CSUMR(11,IPP)                                   DS18
            CSUMR(11,IPP)= ( PSUB(X1) - PSUB(X2) )*100.0                       DS18
      C                                                                        DS18
      C          EQ. 183                                                       DS18
            VEPC= ( SOPERC - (DSET17(8)**2)*FNT )/ FNT1                        DS18
            ARG(1)= 4HVEPC                                                     DS18
            IARG(2)= 183                                                       DS18
            Y= YSUB(VEPC,DSET17(8))                                            DS18
            X1= ( DSET17(7) - 0.9*DSET17(6) )/ Y                               DS18
            X2= ( DSET17(7) - 1.1*DSET17(6) )/ Y                               DS18
      C          STORE CLPRD IN CSUMR(15,IPP)                                  DS18
            CSUMR(15,IPP)= ( PSUB(X1) - PSUB(X2) )*100.0                       DS18
      C                                                                        DS18
        900 RETURN                                                             DS18
            END                                                                DS18
```

```
 FOR,IS DS456
      SUBROUTINE DS456                                                    DS456
C         PROCESSES DATA SETS 4, 5, AND 6 AT THE STRATA LEVEL             DS456
C                                                                         DS456
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                       CASCM
      COMMON  /CASCM /                                                    CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)   ,PPRUTS(5,2)   ,YPRUTS(3,2)  CASCM
     2   ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)     CASCM
     3   ,WPRIOR(4)    ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE          CASCM
C                                                                         CASCM
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                            CASFLG
      COMMON  /CASFLG/                                                    CASFLG
     1   H      ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2   ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3   ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                        CASFLG
     4   ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5   ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR          CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                    CASFLG
C                                                                         CASFLG
C         CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                     DSET4
      COMMON  /DSET4 /                                                    DSET4
     1   STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                  JULY76
     2   ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                     JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4   ,XYS   ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS  ,ANVARS                JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                     JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                            JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )     DSET4
C                                                                         DSET4
C         CAS DATA SET 7 (AT ZONE LEVEL)                                  DSET7
      COMMON  /DSET7 /                                                    DSET7
     1   ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3   JULY76
     2   ,ESTVZ ,HWAZ12                                                   JULY76
     3   ,M1K2CL(10)    ,EPWCL(10)     ,EPW2CL(10)    ,PKPICL(10)         JULY76
     4   ,PK2CL(10)     ,PKCL(10)      ,SSQ(10)                           JULY76
      INTEGER  ZONE                                                       JULY76
      REAL  M1K2KZ, M1K2CL                                                JULY76
      DIMENSION  DSET7(81)                                                JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                         DSET7
```

```
C                                                                           DSET7
C         CAS DATA SET 10 (STRATA DATA -- FINAL PASS)                       JULY76
      COMMON  /DSET10/                                                      JULY76
     1    HWAS   ,TWAS   ,EWAS   ,AERRS ,AVARS ,TPRODS,EPRODS,PRERRS,PRVARS JULY76
     2   ,YS     ,ESTYS  ,YERRS  ,M1JS   ,M2JS   ,CT1S   ,CT2S   ,CT3S   ,ANAVS  JULY76
     3   ,ANPRVS,ES                                                         JULY76
      REAL   M1JS , M2JS                                                    JULY76
      DIMENSION  DSET10(20)                                                 JULY76
      EQUIVALENCE  ( DSET10,HWAS )                                          JULY76
C                                                                           DSET10
C         CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                         JULY76
      COMMON  /DSET11/                                                      DSET11
     1    HWAZ   ,TWAZ   ,EWAZ   ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2   ,TYZ    ,EYZ    ,YERRZ ,M1Z   ,M2Z    ,CT1Z   ,CT2Z   ,CT3Z   ,ANAVZ DSET11
     3   ,ANPRVZ                                                            DSET11
      REAL   M1Z , M2Z                                                      DSET11
      DIMENSION  DSET11(19)                                                 DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                          DSET11
C                                                                           DSET11
C         FILE DEFINITIONS AND RECORD LENGTHS                               FILES
      COMMON  /FILES /                                                      FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ              FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO            FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD            FILES
     4   ,INP    ,OUTP   ,TACQ   ,LTACQ ,CASDSF,LCASDS                       FILES
      INTEGER   SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT   FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF                  FILES
C                                                                           FILES
C         INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)          IXCDSF
      COMMON  /IXCDSF/                                                      IXCDSF
     1    IXCDSF(1),LIXCDS
C                                                                           IXCDSF
C        DEBUGGING PRINT FLAG                                               DS456
      COMMON  /DEBUGF/  DEBUGF                                              DS456
C                                                                           DS456
C                                                                           DS456
  100    CONTINUE                                                           DS456
C         WERE THERE ANY GROUP II SEGMENTS ACQUIRED FOR THIS STRATUM        DS456
      IF ( XM2JS .NE. 0.0 ) GO TO 480                                       DS456
C         NO. RECLASSIFY ALL GROUP II SUBSTRATA AS GROUP III SUBSTRATA      DS456
C         BY ADDING DATA SET TO DATA SET 6, THEN ZEROING OUT DATA SET 5     DS456
```

```
      TWAS3= TWAS3 + TWAS2                                                 DS456
      HWAS3= HWAS3 + HWAS2                                                 DS456
      XCT3S = XCT3S  + XCT2S                                               DS456
      DO 470 I=1,7                                                         JULY76
      DSET5(I)= 0.0                                                        DS456
  470 CONTINUE                                                             DS456
      GO TO 500                                                            DS456
C                                                                          DS456
C        GENERATE REST OF DATA SET 5  (EQNS. 33,36,40, AND 39)             DS456
  480 EWAS2= HWAS2*P2IDPK/XM2JS                                            DS456
      ANVS2= ANVS2/XM2JS                                                   DS456
      IF ( TOPT .NE. 0 .AND. XCT2S .GT. 1.0 ) CALL TSUB                    DS456
C                                                                          DS456
C        WRITE DATA SETS 4,5,6 ON INTERMEDIATE FILE                        DS456
  500 IRSTR= IRSTR + 1                                                     DS456
      NSTRAZ= NSTRAZ + 1                                                   DS456
      NSTRAT= NSTRAT + 1                                                   DS456
C        MOVE YS AND ESTYS FROM DATA DET 10 TO DATA SETS 4,5,6             DS456
C        BEFORE WRITING STRATA DATA ONTO CAS INTERMEDIATE FILE.            DS456
      XYS   = YS                                                           DS456
      XESTYS= ESTYS                                                        DS456
      CALL RANACF (CASDSF,IRSTR,DSET4,LCASDS,IXCDSF,LIXCDS,2)              DS456
C                                                                          DS456
C        AGGREGATE STRATA DATA SETS 4,5,6 UP TO DATA SET 7 (ZONE LEVEL)    DS456
C        ( EQS. 50-56,58,59,61,62,68,69 )                                  DS456
      TWAZ= TWAZ + TWAS1 + TWAS2 + TWAS3                                   DS456
      IF ( XM1JS + XM2JS .EQ. 0.0 )  GO TO 990                             DS456
      M1K2KZ= M1K2KZ + XM1JS   + XM2JS                                     DS456
      HWAZ1= HWAZ1  + HWAS1  + HWAS2                                       DS456
      EWAZ1= EWAZ1  + EWAS1  + EWAS2                                       DS456
C                                                                          DS456
  990 RETURN                                                               JULY76
      END                                                                  DS456
```

```
FOR,IS DS7
      SUBROUTINE DS7                                                    DS7
C                                                                       DS7
C          PROCESSES DATA SET 7 AT THE ZONE LEVEL.                      DS7
C                                                                       DS7
C          ARGUMENT LIST FOR ERROR PROCESSING                           ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1    NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C          CAS CONTROL CARD INPUT DATA AND CONSTANTS                    CASCM
      COMMON  /CASCM /                                                  CASCM
     1    AREACF,YCF   ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)      CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C          FLAGS AND COUNTERS FOR CAS SIMULATOR                         CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1    H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS        CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR        CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C          CONSTANT QUANTITIES FOR LEM PROGRAM                          CONST
      COMMON  /CONST /                                                  CONST
     1    NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                      CONST
C                                                                       CONST
      COMMON/FILES1/                                                    FILES1
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C          CAS DATA SET 7 (AT ZONE LEVEL)                               DSET7
      COMMON  /DSET7 /                                                  DSET7
     1    ZONE  ,HWAZ2 ,EZ    ,M1K2KZ,ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3 JULY76
     2  ,ESTVZ ,HWAZ12                                                  JULY76
     3  ,M1K2CL(10)   ,EPWCL(10)    ,EPW2CL(10)   ,PKPICL(10)           JULY76
     4  ,PK2CL(10)    ,PKCL(10)     ,SSQ(10)                            JULY76
      INTEGER  ZONE                                                     JULY76
      REAL  M1K2KZ, M1K2CL                                              JULY76
```

```
      DIMENSION  DSET7(81)                                              JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                       DSET7
C                                                                       DSET7
C        CAS DATA SET 8 (AT REGION LEVEL)                               DSET8
      COMMON  /DSET8 /                                                  DSET8
     1   REGION,HWAR2 ,ER     ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR  JULY76
     2  ,M1M2ZR,FILL8(71)                                               JULY76
      INTEGER  REGION                                                   JULY76
      REAL  M1K2KR                                                      JULY76
      DIMENSION  DSET8(10)                                              JULY76
      EQUIVALENCE  ( DSET8,REGION )                                     DSET8
C                                                                       DSET8
C        CAS DATA SET 11 (ZONE DATA -- FINAL PASS)                      JULY76
      COMMON  /DSET11/                                                  DSET11
     1   HWAZ  ,TWAZ  ,EWAZ  ,AERRZ ,AVARZ ,TPRODZ,EPRODZ,PRERRZ,PRVARZ DSET11
     2  ,TYZ   ,EYZ   ,YERRZ ,M1Z   ,M2Z   ,CT1Z  ,CT2Z  ,CT3Z  ,ANAVZ  DSET11
     3  ,ANPRVZ                                                         DSET11
      REAL  M1Z , M2Z                                                   DSET11
      DIMENSION  DSET11(19)                                             DSET11
      EQUIVALENCE  ( DSET11,HWAZ )                                      DSET11
C                                                                       DSET11
C        CAS DATA SET 12 (REGION DATA -- FINAL PASS)                    JULY76
      COMMON  /DSET12/                                                  DSET12
     1   HWAR  ,TWAR  ,EWAR  ,AERRR ,AVARR ,TPRODR,EPRODR,PRERRR,PRVARR DSET12
     2  ,TYR   ,EYR   ,YERRR ,M1R   ,M2R   ,CT1R  ,CT2R  ,CT3R  ,ANAVR  DSET12
     3  ,ANPRVR                                                         DSET12
      REAL  M1R , M2R                                                   DSET12
      DIMENSION  DSET12(19)                                             DSET12
      EQUIVALENCE  ( DSET12,HWAR )                                      DSET12
C                                                                       DSET12
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        INDEX RECORD FOR CAS INTERMEDIATE DATA SET FILE (CASDSF)       IXCDSF
      COMMON  /IXCDSF/                                                  IXCDSF
```

```
     1    IXCDSF(1),LIXCDS
C                                                                       IXCDSF
C       DEBUGGING PRINT FLAG                                            DS7
      COMMON  /DEBUGF/  DEBUGF                                          DS7
C                                                                       DS7
      DOUBLE PRECISION XGOP,ZER
      DOUBLE PRECISION A,B,D,      XG01,XG02,XG03,XG04,XG05,XG06
      DATA XGOP/2.0D0/,ZER/0.0D0/
C                                                                       DS7
C     LOCAL VARIABLES                                                   DS7
C        A     = REGRESSION COEFFICIENT USED TO CALCULATE  S**2         DS7
C        B     = REGRESSION COEFFICIENT USED TO CALCULATE  S**2         DS7
C        D     = DENOMINATOR OF B                                       DS7
C        SOSQ  = SO**2                                                  DS7
C                                                                       DS7
100     CONTINUE                                                        DS7
C                                                                       DS7
C                                                                       DS7
C        GENERATE REST OF DATA SET 7 (ZONE LEVEL)                       DS7
C                                                                       DS7
C     IF M1K2KZ = 0, NO ACQUIRED GROUP I OR GROUP II SEGMENTS IN ZONE.  DS7
C        HWAZ2 = EZ = ANALVZ = HWAZ1 = EWAZ1 = SSQ(CLASS) = 0.0         JULY76
      IF ( M1K2KZ .EQ. 0.0 )  GO TO 780                                 JULY76
C                                                                       DS7
C        M1K2KZ .GT. 0. COMPUTE EZ (EQ. 63)                             DS7
      IF ( HWAZ1 .NE. 0.0 )  EZ= EWAZ1/HWAZ1                            DS7
C        IF M1K2KZ .LT. 2, THEN HWAZ2 = 0 AND SSQ(CLASS) = 0 FOR ALL    JULY76
C        CLASSES IN ZONE                                                JULY76
      IF ( M1K2KZ .LT. 2.0 )  GO TO 780                                 JULY76
C                                                                       DS7
C        M1K2KZ .GE. 2. SET HWAZ2 = HWAZ1 (EQ. 57)                      JULY76
 740  HWAZ2= HWAZ1                                                      DS7
      M1M2ZR= 1                                                         DS7
C                                                                       DS7
C        COMPUTE S**2  ( EQS. 64-67 )                                   DS7
C        FOR EACH CLASS IN ZONE                                         JULY76
      DO 770 ICL=1,MXCLSS                                               JULY76
      IF ( M1K2CL(ICL) .EQ. 0.0 )  GO TO 780                            JULY76
      IF(M1K2CL(ICL) .GT. 1.0)GO TO 750
C        LESS THAN 2 ACQUIRED SEGMENTS IN CLASS.                        JULY76
C        ON THE FIRST MONTE CARLO ITERATION PRINT WARNING AND CONTINUE  JULY76
```

```
      IARG(1)= ICL                                                      JULY76
      IARG(2)= M1K2CL(ICL)                                              JULY76
      CALL ERRMES (3HCAS,3HDS7,19,0)                                    JULY76
      GO TO 770                                                         JULY76
C                                                                       JULY76
 750  IF ( M1K2CL(ICL) .LT. H )  GO TO 760                              JULY76
C         NUMBER OF ACQUIRED SEGMENTS IN CLASS .GE. H.                  JULY76
C         USE REGRESSION FORMULA                                        JULY76
      XG01 = M1K2CL(ICL)
      XG02 = PK2CL(ICL)
      XG03 = PKCL(ICL)
      XG04 = PKPICL(ICL)
      XG05 = EPWCL(ICL)
      XG06 = EPW2CL(ICL)
      D = XG01*XG02 -XG03**2
      IF(D .EQ. ZER)GO TO 760
      B = (XG01*XG04 - XG05*XG03)/D
      A = (XG05 - B*XG03)/XG01
      SOSQ = (XG06 - A*XG05 - B*XG04)/(XG01 - XGOP)
      GO TO 765                                                         JULY76
C                                                                       JULY76
C         LESS THAN H BUT MORE THAN 1 ACQUIRED SEGMENTS IN ZONE.        JULY76
C         USE VARIANCE FORMULA.                                         JULY76
 760  SOSQ = ( EPW2CL(ICL) - EPWCL(ICL)**2 / M1K2CL(ICL) ) /            JULY76
     1  ( M1K2CL(ICL) - 1.0 )                                           JULY76
C                                                                       JULY76
 765  SSQ(ICL) = AMIN1(SOSQ,S2MAX)                                      JULY76
C                                                                       JULY76
 770  CONTINUE                                                          JULY76
C                                                                       DS7
C         WRITE DATA SET 7 ONTO INTERMEDIATE FILE                       DS7
 780  IRZONE= IRZONE + 1                                                DS7
      NZTOT= NZTOT + 1                                                  DS7
      NZONES= NZONES + 1                                                DS7
      CALL RANACF (CASDSF,IRZONE,DSET7,LCASDS,IXCDSF,LIXCDS,2)          DS7
C                                                                       DS7
C         AGGREGATE ZONE DATA SET 7 UP TO DATA SET 8 (REGION LEVEL)     DS7
C         EQNS. 70-76,78,79                                             DS7
      TWAR  = TWAR   + TWAZ                                             DS7
      HWAR2 = HWAR2  + HWAZ2                                            DS7
      M1K2KR= M1K2KR + M1K2KZ                                           DS7
```

```
      HWAR1 = HWAR1  + HWAZ1                                          DS7
      EWAR1 = EWAR1  + EWAZ1                                          DS7
C                                                                     DS7
990   RETURN                                                          DS7
      END                                                             DS7
```

@ ELT EJECT,1,760427, 39011   , 1

```
      SUBROUTINE EJECT (NLINES)                                         EJECT
C        RESTORES PAGE AND PRINTS THE PAGE HEADER                       EJECT
C                                                                       EJECT
C        COMMON BLOCK DEFINITIONS                                       EJECT
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,TYPS  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C                                                                       EJECT
      NLINE= NLINES + 2                                                 EJECT
      NPAGE= NPAGE + 1                                                  EJECT
      WRITE (OUTP,1)  TITLE, ICASE, NPAGE                               EJECT
    1 FORMAT (1H1,4X,10A6,5X,23HLPP SIMULATION     CASE,I5,7H   PAGE,I4 EJECT
     1  /)                                                              EJECT
      IF ( NSTTL .EQ. 0 )  GO TO 900                                    EJECT
      WRITE (OUTP,2)  SUBTTL                                            EJECT
      NLINE= NLINE + 3                                                  EJECT
    2 FORMAT (/25X,10A6/)                                               EJECT
  900 RETURN                                                            EJECT
      END                                                               EJECT
```

@ ELT ERRMC,1,760427, 39012 , 1

```
      SUBROUTINE ERRMC                                                  ERRMC
C        ERROR MODEL CONTROL ROUTINE FOR THE LEM PROGRAM                ERRMC
C        FOR EACH MONTE CARLO ITERATION ERRMC PROPERLY INITIALIZES THE  ERRMC
C        RANDOM NUMBER SEEDS FOR THE FOLLOWING ERROR SOURCES ..         ERRMC
C           * CLASSIFICATION ERROR (ICLASS)                             ERRMC
C           * SIGNATURE EXTENSION ERROR (ISEXT)                         ERRMC
C           * SEGMENT CROP CALENDAR ERROR (ISCC)                        ERRMC
C           * CAS GROUP II NON-EPOCH YEAR, HISTORICAL PROPORTION WHEAT  ERRMC
C             (ICAS2)                                                   ERRMC
C           * CAS GROUP III MULTI-YEAR PROPORTION OF WHEAT (ICAS3)      ERRMC
C                                                                       ERRMC
C        COMMON BLOCK DEFINITIONS                                       ERRMC
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASF  ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR   ,ENDZ   ,ISTG   ,ICAMS  ,IYES   ,IAGO   ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSFSG,ICSECH,ICSFSH,ICSECE LEMCM
     4  ,ICSFYM,ICSFSF,ICSLAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C                                                                       ERRMC
C        INPUTS ...   ICAMS, ICLASS, ISEXT, ISCC, ICAS2, ICAS3, AND RSEEDERRMC
C                                                                       ERRMC
C        OUTPUTS ...    SEED                                            ERRMC
C                                                                       ERRMC
C        LINKAGE ...    CALL ERRMC                                      ERRMC
C          ERRMC IS CALLED FROM THE LEM DRIVER                          ERRMC
C                                                                       ERRMC
C ********************************************************************* ERRMC
C                                                                       ERRMC
C                                                                       ERRMC
C                                                                       ERRMC
C        TEST THE CAMS OPTION. IF ICAMS .NE. 0, ALL 3 CAMS ERRORS ARE   ERRMC
C        CONSTANT AND CAMS WILL BE CALLED ONLY ONCE.                    ERRMC
C        IF ICAMS = 0, THEN IN GENERAL EACH OF THE CAMS ERRORS MAY BE   ERRMC
C        VARIED OR HELD CONSTANT INDEPENDENTLY.                         ERRMC
C                                                                       ERRMC
      IF ( ICAMS .NE. 0 )  GO TO 200                                    ERRMC
C        ICAMS = 0. VARY CAMS ERRORS OR HOLD THEM CONSTANT AS SPECIFIED ERRMC
C        BY THE PARAMETERS  ICLASS, ISEXT, AND ISCC                     ERRMC
      IF ( ICLASS .EQ. 1 )   SEED(2)= RSEED(2)                          ERRMC
      IF ( ISEXT  .EQ. 1 )   SEED(3)= RSEED(3)                          ERRMC
      IF ( ISCC   .EQ. 1 )   SEED(4)= RSEED(4)                          ERRMC
C                                                                       ERRMC
  200 IF ( ICAS2  .EQ. 1 )   SEED(6)= RSEED(6)                          ERRMC
      IF ( ICAS3  .EQ. 1 )   SEED(7)= RSEED(7)                          ERRMC
  900 RETURN                                                            ERRMC
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE ERRMES (PROG,SUBR,ICODE,LEVEL)                          ERRMES
C        CONTROLS THE PRINTING OF ALL ERROR MESSAGES FOR LEM.            ERRMES
C                                                                        ERRMES
C        INPUT PARAMETERS ...                                            ERRMES
C        PROG = SUBPROGRAM NAME IN A6 FORMAT (E.G. 3HLEM,4HCAMS, ETC.)   ERRMES
C        SUBR = SUBROUTINE NAME IN A6 FORMAT (E.G. 5HINPUT)              ERRMES
C        ICODE= ERROR CODE                                               ERRMES
C        LEVEL= ERROR SEVERITY LEVEL (= 0 FOR NONFATAL, = 1 FOR FATAL)   ERRMES
C                                                                        ERRMES
C                                                                        ERRMES
C        PARAMETERS TO BE PRINTED AS PART OF ERROR MESSAGE ARE PASSED    ERRMES
C        IN THE ARRAY ARG IN /ARGLST/                                    ERRMES
C                                                                        ERRMES
C        COMMON BLOCK DEFINITIONS                                        ERRMES
C        ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                             ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
C        CONSTANT QUANTITIES FOR LEM PROGRAM                             CONST
      COMMON  /CONST /                                                   CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSEG                        CONST
C                                                                        CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                   FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4   ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT   FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                 FILES
C                                                                        FILES
C                                                                        ERRMES
C        LOCAL VARIABLES                                                 ERRMES
C     IMES  = ERROR MESSAGE CODE                                         ERRMES
C     BLANK = 3H    = WORD OF BLANKS USED TO FILL PART OF ERROR MESSAGE. ERRMES
C     NONFTL= 3HNON = PART OF ERROR MESSAGE                              ERRMES
C                    (DISTINGUISHES BETWEEN NONFATAL AND FATAL ERRORS)   ERRMES
C     ERRLVL= BLANK OR NONFTL (USED TO FILL PART OF ERROR MESSAGE)       ERRMES
C                                                                        ERRMES
      REAL  NONFTL                                                       ERRMES
C                                                                        ERRMES
      DATA  BLANK,NONFTL                                                 ERRMES
     1   /  1H   ,3HNON /                                                ERRMES
C                                                                        ERRMES
C        LINKAGE ... CALL ERRMES (PROG,SUBR,ICODE,LEVEL)                 ERRMES
C          ERRMES IS CALLED FROM LEM, INPUT, INPCHK                      ERRMES
C                                                                        ERRMES
C         SUBROUTINES USED ...   INPERR                                  ERRMES
C                                                                        ERRMES
C ********************************************************************** ERRMES
C                                                                        ERRMES
      IMES= ICODE                                                        ERRMES
      IF ( IMES .EQ. 99 )  GO TO 990                                     ERRMES
C                                                                        ERRMES
      IF ( LEVEL .NE. 0 )  GO TO 20                                      ERRMES
C        NONFATAL ERROR                                                  ERRMES
```

```
          ERRLVL= NONFTL                                                   ERRMES
          NERRS= NERRS + 1                                                 ERRMES
          GO TO 30                                                         ERRMES
C            FATAL ERROR                                                   ERRMES
   20     ERRLVL= BLANK                                                    ERRMES
          NFATAL= NFATAL + 1                                               ERRMES
C                                                                          ERRMES
   30     CALL PAGER (6)                                                   ERRMES
          WRITE (OUTP,1000)  ERRLVL,PROG,SUBR,IMES                         ERRMES
 1000 FORMAT (// 6H *****,A3,26HFATAL ERROR IN SUBPROGRAM ,A6,             ERRMES
     1  14H, SUBROUTINE ,A6,13H  ERROR CODE ,I3,7H  *****)                 ERRMES
C                                                                          ERRMES
C            TEST FOR CALLING SUBPROGRAM/ROUTINE AND CALL PROPER LOWER LEVELERRMES
C            ROUTINE TO PRINT ERROR MESSAGE                                ERRMES
          IF ( SUBR .EQ. 5HINPUT )  GO TO 150                              ERRMES
          IF ( SUBR .NE. 6HINPCHK )  GO TO 200                             ERRMES
C            ERROR DETECTED IN LEM INPUT PROCESSOR                         ERRMES
  150     CALL INPERR (IMES)                                               ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
  200     IF ( SUBR .NE. 6HCAMSIN )  GO TO 300                             ERRMES
C                                                                          ERRMES
C            ERROR DETECTED IN CAMS CONTROL CARD DATA                      ERRMES
          CALL CAMERS (IMES)                                               ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
  300     IF ( SUBR .NE. 5HCASIN )  GO TO 400                              ERRMES
          IF ( IMES .GT. 17 )  GO TO 150                                   ERRMES
C                                                                          ERRMES
C            ERROR DETECTED IN CAS CONTROL CARD DATA                       ERRMES
          CALL CASER1 (IMES)                                               ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
  400     NPERRS= NPERRS + 1                                               ERRMES
          IF ( PROG .NE. 4HCAMS )  GO TO 500                               ERRMES
C                                                                          ERRMES
C            ERROR DETECTED IN CAMS MODULE                                 ERRMES
          CALL CAMER2 (IMES)                                               ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
  500     IF ( PROG .NE. 3HYES )  GO TO 600                                ERRMES
C                                                                          ERRMES
          CALL PAGER (3)                                                   ERRMES
C            ERROR DETECTED IN MODULE YES.                                 ERRMES
          GO TO (510,520), IMES                                            ERRMES
  510     WRITE (OUTP,921)                                                 ERRMES
  921     FORMAT (/62H0YES INPUT FILE (YESERR) - BEGINNING REGION AND ZONE NERRMES
         1OT FOUND )                                                       ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
  520     WRITE (OUTP,922)                                                 ERRMES
  922     FORMAT (/59H0YES INPUT FILE (YESERR) - ENDING REGION AND ZONE NOT ERRMES
         1FOUND )                                                          ERRMES
C                                                                          ERRMES
  600     IF ( PROG .NE. 3HCAS )  GO TO 700                                ERRMES
C                                                                          ERRMES
C            ERROR DETECTED IN CAS SIMULATOR MODULE                        ERRMES
          CALL CASER2 (IMES)                                               ERRMES
          GO TO 900                                                        ERRMES
C                                                                          ERRMES
```

```
 700  IF ( PROG .NE. 3HSTG )  GO TO 900                                  ERRMES
      CALL STGERR (IMES)                                                  ERRMES
C                                                                         ERRMES
 900  RETURN                                                              ERRMES
 990  CALL PAGER (3)                                                      ERRMES
      WRITE (OUTP,1001)                                                   ERRMES
1001  FORMAT (//45H JOB TERMINATED IN ERRMES DUE TO FATAL ERRORS )        ERRMES
      CALL WRAPUP                                                         ERRMES
      STOP                                                                ERRMES
      END                                                                 ERRMES
```

@ ELT FZULU,1,760428, 52543 , 1

```
      SUBROUTINE FZULU(IDATE,IOUT)                                      FZULU
C ***********************************************************************FZULU
C                                                                       FZULU
C ROUTINE       FZULU                                                   FZULU
C                                                                       FZULU
C PURPOSE       TO CONVERT ZULU DATE TO YEAR MONTH AND DAY              FZULU
C                                                                       FZULU
C LINKAGE       CALL FZULU (IDATE,IOUT)                                 FZULU
C                   IDATE,  BINARY INTEGER ZULU DATE                    FZULU
C                   IOUT,   CALENDER DATE,DIMENSION 3  FOR INTEGER      FZULU
C                           YEAR, MONTH AND DAY RESPECTIVELY            FZULU
C                                                                       FZULU
C ROUTINES CALLED   NONE                                                FZULU
C                                                                       FZULU
C LOCAL VARIABLES:                                                      FZULU
C                                                                       FZULU
C                   IDAYS     VECTOR CONTAINING NO. DAYS OF YEAR ON     FZULU
C                             LAST DAY OF MONTHS 0 THROUGH 12 FOR NORMALFZULU
C                             YEAR FOLLOWED BY 13 MONTHS OF LEAP YEAR   FZULU
C                                                                       FZULU
C                   NLEAP,    NO. OF LEAP YEARS SINCE 1900 TO SPECIFIED FZULU
C                             DATE                                      FZULU
C                   NFLG,     SUBSCRIPT INTO IDAYS VECTOR               FZULU
C                                       START=1, NOT LEAP YEAR          FZULU
C                                       START=14, LEAP YEAR             FZULU
C                   JDAY,     JULIAN DAY OF YEAR                        FZULU
C                   I,        LOCAL USE                                 FZULU
C COMMENTS          NONE                                                FZULU
C***********************************************************************FZULU
      DIMENSION IOUT (3)                                                FZULU
      DIMENSION IDAYS(26)                                               FZULU
      DATA IDAYS/0,31,59,90,120,151,181,212,243,273,304,334,365,        FZULU
     1           0,31,60,91,121,152,182,213,244,274,305,335,366/        FZULU
      IDATE = IDATE + 18263                                             FZULU
      DO 5 I=1,3                                                        FZULU
      IOUT(I) = 0                                                       FZULU
    5 CONTINUE                                                          FZULU
C               FIND NO. LEAP YEARS SINCE 1900 AND GET CURRENT YEAR     FZULU
      NLEAP=(1095+IDATE)/1461                                           FZULU
      IOUT(1) = (IDATE-NLEAP)/365
C               SEE IF CURRENT YEAR ISLEAP YEAR-SET NFLG=1 OR 14        FZULU
      I=IOUT(1)/4
      I=I*4                                                             FZULU
      NFLG=1                                                            FZULU
      IF(IOUT(1) - I)10,10,20                                                 *NEW
   10 NFLG=14                                                           FZULU **-1
C               JULIAN DAY = ZULU DAY-NO YEARS*365-NO.LEAP YEARS        FZULU
   20 JDAY=IDATE-(IOUT(1)*365)-NLEAP + 1
C               GET DAY, MONTH FROM TABLE SEARCH                        FZULU
C                                                                       FZULU
   30 IF(JDAY-IDAYS(NFLG))50,50,40                                      FZULU
   40 IOUT(2)=IOUT(2)+1                                                 FZULU
      NFLG=NFLG+1                                                       FZULU
      GOTO 30                                                           FZULU
   50 NFLG=NFLG-1                                                       FZULU
      IOUT(3)=JDAY-IDAYS(NFLG)                                          FZULU
      RETURN                                                            FZULU
      END                                                               FZULU
```

```
FOR,IS GETYS
      SUBROUTINE GETYS                                                  GETYS
C         READS STRATA YIELD DATA FROM YESOUT FILE AND OBTAINS THE      GETYS
C         PROPER VALUE OF ESTIMATED YIELD FOR THE CURRENT BIOWINDOW     GETYS
C         OR PREDICTION DATE.                                           GETYS
C                                                                       GETYS
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                          CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C         CONTROL PARAMETERS FOR LEM PROGRAM                            CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C         CONSTANT QUANTITIES FOR LEM PROGRAM                           CONST
      COMMON  /CONST /                                                  CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C         CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                   DSET4
      COMMON  /DSET4 /                                                  DSET4
     1   STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                JULY76
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                    JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS   ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS  ,ANVARS               JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                   JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                          JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )   DSET4
```

28234-6029-RU-51
Page 455

```
C                                                                         DSET4
C         CAS DATA SET 7 (AT ZONE LEVEL)                                  DSET7
      COMMON  /DSET7 /                                                    DSET7
     1    ZONE  ,HWAZ2 ,EZ    ,M1K2KZ.ANALVZ,NSTRAZ,HWAZ1 ,EWAZ1 ,HWAZ3   JULY76
     2   ,ESTVZ ,HWAZ12                                                   JULY76
     3   ,M1K2CL(10)    ,EPWCL(10)     ,EPW2CL(10)    ,PKPICL(10)         JULY76
     4   ,PK2CL(10)     ,PKCL(10)      ,SSQ(10)                           JULY76
      INTEGER  ZONE                                                       JULY76
      REAL  M1K2KZ, M1K2CL                                                JULY76
      DIMENSION. DSET7(81)                                                JULY76
      EQUIVALENCE  ( DSET7,ZONE )                                         DSET7
C                                                                         DSET7
C         CAS DATA SET 8 (AT REGION LEVEL)                                DSET8
      COMMON  /DSET8 /                                                    DSET8
     1    REGION,HWAR2 ,ER    ,M1K2KR,ANALVR,NZONES,HWAR1 ,EWAR1 ,ESTVR   JULY76
     2   ,M1M2ZR,FILL8(71)                                                JULY76
      INTEGER  REGION                                                     JULY76
      REAL  M1K2KR                                                        JULY76
      DIMENSION  DSET8(10)                                                JULY76
      EQUIVALENCE  ( DSET8,REGION )                                       DSET8
C                                                                         DSET8
C         CAS DATA SET 10 (STRATA DATA -- FINAL PASS)                     JULY76
      COMMON  /DSET10/                                                    JULY76
     1    HWAS  ,TWAS  ,EWAS  ,AERRS ,AVARS ,TPRODS,EPRODS,PRERRS,PRVARS  JULY76
     2   ;YS    ,ESTYS ,YERRS ,M1JS  ,M2JS  ,CT1S  ,CT2S  ,CT3S  ,ANAVS   JULY76
     3   ,ANPRVS,ES                                                       JULY76
      REAL  M1JS , M2JS                                                   JULY76
      DIMENSION  DSET10(20)                                               JULY76
      EQUIVALENCE  ( DSET10,HWAS )                                        JULY76
C                                                                         DSET10
C         FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                    FILES
     1    SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO          FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
     4   ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                       FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT    FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                  FILES
C                                                                         FILES
C         SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                      SSHDTA
      COMMON  /SSHDTA/                                                    SSHDTA
```

```
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                              JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                    JULY76
      DIMENSION  SSHDTA(39)                                             JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                    SSHDTA
C                                                                       SSHDTA
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C        YIELD DATA FROM YESOUT FILE                                    YESDTA
      COMMON  /YESDTA/                                                  YESDTA
     1   YSTR  ,IZPRDD(6)    ,YSCI(6)       ,VSYCI(6)                   YESDTA
     2  ,RDYES ,NYESPP                                                  YESDTA
      INTEGER  RDYES                                                    YESDTA
C                                                                       YESDTA
C       DEBUGGING PRINT FLAG                                            GETYS
      COMMON  /DEBUGF/  DEBUGF                                          GETYS
C                                                                       GETYS
C                                                                       GETYS
C        READ STRATA YIELD DATA FROM YESOUT FILE                        GETYS
      NRYES= NRYES + 1                                                  GETYS
      READ (YESOUT)  YCOUN,REGION,ZONE,STRATA,YSTR                      GETYS
     1  , ( IZPRDD(I),YSCI(I),VSYCI(I),I=1,6 )                          GETYS
      IF ( YCOUN .EQ. ENDFIL )  GO TO 999                               GETYS
C                                                                       GETYS
C        ON THE FIRST ITERATION OF THIS RUN, CHECK FOR CONSISTENCY      GETYS
C        BETWEEN YESOUT AND SUBHST FILES.                               GETYS
      IF ( NT .GT. NSTART )  GO TO 110                                  GETYS
      IF ( NSTRAT .EQ. 0 )  GO TO 110                                   GETYS
      IF ( YCOUN .EQ. ENDFIL )  GO TO 999                               GETYS
      IF ( REGION .NE. IREG2 )  GO TO 999                               GETYS
      IF ( ZONE .NE. IZONE2 )  GO TO 999                                GETYS
      IF ( STRATA .NE. ISTRA2 )  GO TO 999                              GETYS
C                                                                       GETYS
 110  YS= YSTR                                                          GETYS
      IF ( PPFLG .NE. 0 )  GO TO 200                                    GETYS
C                                                                       GETYS
C        FIND LAST NONZERO YIELD DATE FOR THIS STRATA                   GETYS
```

```
      NYESPP= 6                                                          GETYS
      DO 120 I=1,6                                                       GETYS
      IF ( IZPRDD(NYESPP) .GT. 0 )  GO TO 130                            GETYS
  120 NYESPP= NYESPP - 1                                                 GETYS
C                                                                        GETYS
C        ALL YIELD DATES ARE ZERO FOR THIS STRATA                        GETYS
      CALL ERRMES (3HCAS,6HCASER2,7,1)                                   GETYS
      YSTR= -1.0                                                         GETYS
      GO TO 900                                                          GETYS
C                                                                        GETYS
C        PICK UP LAST VALUE OF ESTIMATED YIELD AND YIELD VARIANCE FOR    GETYS
C        BIOWINDOW (IBW)                                                 GETYS
  130 ESTYS= YSCI(NYESPP)                                                GETYS
      EVYRS= VSYCI(NYESPP)**2                                            GETYS
C                                                                        GETYS
C                                                                        GETYS
C           TEMPORARY DEBUGGING PRINTOUT                                 GETYS
      GO TO 900                                                          GETYS
C                                                                        GETYS
C        PICK UP ESTIMATED YIELD AND YIELD VARIANCE FOR PREDICTION DATE  GETYS
  200 II= 6                                                              GETYS
      DO 210 I=1,6                                                       GETYS
      IF ( IZPRDD(II) .EQ. 0 )  GO TO 210                                GETYS
      IF ( PPDATE .GE. IZPRDD(II) )  GO TO 220                           GETYS
  210 II= II - 1                                                         GETYS
C        ERROR.  PREDICTION DATE PPDATE .LT. ALL ZULU PREDICTION DATES   GETYS
C        ON YESOUT FILE.                                                 GETYS
      IARG(1)= IPD                                                       GETYS
      IARG(2)= PPDATE                                                    GETYS
      CALL ERRMES (3HCAS,5HGETYS,16,0)                                   GETYS
      YSTR= -1.0                                                         GETYS
      GO TO 900                                                          GETYS
C                                                                        GETYS
  220 ESTYS= YSCI(II)                                                    GETYS
      EVYRS= VSYCI(II)**2                                                GETYS
C                                                                        GETYS
C                                                                        GETYS
C           TEMPORARY DEBUGGING PRINTOUT                                 GETYS
C                                                                        GETYS
C                                                                        GETYS
C                                                                        GETYS
```

```
C                                                                       GETYS
C            TEMPORARY DEBUGGING PRINTOUT                               GETYS
  900 CONTINUE
        RETURN                                                          GETYS
C                                                                       GETYS
C                                                                       GETYS
C                                                                       GETYS
  999 CALL ERRMES (3HCAS,5HGETYS,10,1)                                  GETYS
      GO TO 900                                                         GETYS
      END                                                               GETYS
```

```
 FOR,IS GROUP
       SUBROUTINE GROUP                                                 GROUP
C         READS SEGMENT DATA FROM THE CAMS OUTPUT FILE (CAMSF), SELECTS GROUP
C         THE ESTIMATED PROPORTION WHEAT FOR THE PROPER BIOWINDOW FOR   GROUP
C         EACH SEGMENT, AND AGGREGATES THE SEGMENT DATA UP TO THE       GROUP
C         SUBSTRATA LEVEL                                               GROUP
C                                                                       GROUP
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
       COMMON  /ARGLST/                                                 ARGLST
      1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                           ARGLST
       DIMENSION  IARG(10)                                              ARGLST
       EQUIVALENCE  ( IARG,ARG )                                        ARGLST
C                                                                       ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
       COMMON  /CASCM /                                                 CASCM
      1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2) CASCM
      2  ,AREAPS,S2MAX ;NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)     CASCM
      3  ,WPRIOR(4)     ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)         CASCM
       INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE       CASCM
C                                                                       CASCM
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                          CASFLG
       COMMON  /CASFLG/                                                 CASFLG
      1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS CASFLG
      2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS        CASFLG
      3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
      4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13 CASFLG
      5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR        CASFLG
       INTEGER  PPFLG , WINDOW , PPDATE                                 CASFLG
C                                                                       CASFLG
C         CAS DATA SETS 1,2, AND 3                                      DSET1
       COMMON  /DSET1 /                                                 DSET1
      1   ISUBST,TWAK  ,HWAK  ,EWAK  ,M1K   ,CT1K  ,ANALVK,EPWK  ,EPW2K JULY76
      2  ,SMPKPI,SUMPK2,SUMPK ,KSUB  ,NCLASS                            JULY76
       REAL  M1K , M2K                                                  JULY76
       DIMENSION  DSET1(14), DSET2(14), DSET3(6)                        JULY76
       EQUIVALENCE  ( DSET1,DSET2,DSET3,ISUBST )                        DSET1
      1  , ( M2K,M1K ), ( CT2K,CT3K,CT1K )                              DSET1
C                                                                       DSET1
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
       COMMON  /FILES /                                                 FILES
      1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ         FILES
```

```
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                       FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT   FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                 FILES
C                                                                        FILES
C         LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                   LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                       LEMCM
      DIMENSION  RSEED(7)                                                LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5        LEMCM
     1  ,RSEED6,RSEED7                                                   LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                      LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                          LEMCM
C                                                                        LEMCM
C         SEGMENT DATA FROM CAMS OUTPUT FILE (CAMSF)                     SEGDTA
      COMMON  /SEGDTA/                                                   SEGDTA
     1   IDSEGT(5)     ,ISEG  ,TPWKI ,ZACDAY(4)     ,EPWKI(4)            SEGDTA
     2  ,ERRPWI(4)     ,ESTPWI                                           SEGDTA
      INTEGER  ZACDAY                                                    SEGDTA
C                                                                        SEGDTA
C         SUBSTRATA HISTORICAL DATA FROM SUBHST FILE                     SSHDTA
      COMMON  /SSHDTA/                                                   SSHDTA
     1   COUN2 ,IREG2 ,IZONE2,ISTRA2,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW SSHDTA
     2  ,AREAK ,PWK   ,NAGR  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3    SSHDTA
     3  ,CV4   ,VMULTK,CLASS(18),MXK,RDSSH                               JULY76
      INTEGER  GRPNO , CLASS , RDSSH                                     JULY76
      DIMENSION  SSHDTA(39)                                              JULY76
      EQUIVALENCE  ( SSHDTA, COUN2 )                                     SSHDTA
C                                                                        SSHDTA
C        DEBUGGING PRINT FLAG                                            GROUP
      COMMON  /DEBUGF/  DEBUGF                                           GROUP
C                                                                        GROUP
C                                                                        GROUP
      DO 490 N=1,NSEG                                                    GROUP
C         READ SEGMENT DATA FROM CAMSF                                   GROUP
      NRCAMS= NRCAMS + 1                                                 GROUP
```

```
      READ (CAMSF)  IDSEGT,ISEG,TPWKI,(ZACDAY(I),EPWKI(I),ERRPWI(I)     GROUP
     1   ,I=1,4 )                                                       GROUP
C                                                                       GROUP
C                                                                       GROUP
C             TEMPORARY DEBUGGING PRINTOUT                              GROUP
C                                                                       GROUP
C                                                                       GROUP
C           TEST FOR CONSISTENCY BETWEEN CAMSF AND SUBHST               GROUP
      IF ( IDSEGT(2) .NE. IREG2 )  GO TO 120                            GROUP
      IF ( IDSEGT(3) .NE. IZONE2 )  GO TO 120                           GROUP
      IF ( IDSEGT(4) .NE. ISTRA2 )  GO TO 120                           GROUP
      IF ( IDSEGT(5) .EQ. ISUBS2 )  GO TO 200                           GROUP
C           INCONSISTENCY BETWEEN CAMSF AND SUBHST.                     GROUP
C           REGION, ZONE, STRATA, AND/OR SUBSTRATA FROM CAMSF AND SUBHST  GROUP
C           DO NOT AGREE.                                               GROUP
  120 CALL ERRMES (3HCAS,5HGROUP,11,1)                                  GROUP
      GO TO 900                                                         GROUP
C                                                                       GROUP
  200 IF ( PPFLG .EQ. 0 )  GO TO 300                                    GROUP
C                                                                       GROUP
C           PPFLG = 1.  PROCESSING PREDICTION DATE.                     GROUP
      DO 210 I=1,4                                                      GROUP
      WINDOW= WPRIOR(I)                                                 GROUP
      IF ( WINDOW .EQ. 0 )  GO TO 490                                   GROUP
      IF ( ZACDAY(WINDOW) .EQ. 0 )  GO TO 210                           GROUP
      IF ( ZACDAY(WINDOW) .LE. PPDATE )  GO TO 400                      GROUP
  210 CONTINUE                                                          GROUP
      GO TO 490                                                         GROUP
C                                                                       GROUP
C                                                                       GROUP
C           BIOWINDOW BEING PROCESSED                                   GROUP
  300 WINDOW= IBW                                                       GROUP
      IF ( ZACDAY(WINDOW) .EQ. 0 )  GO TO 490                           GROUP
C                                                                       GROUP
C           CONVERT EPWKI FROM PERCENT TO FRACTION                      GROUP
  400 ESTPWI= EPWKI(WINDOW)*0.01                                        GROUP
C           EQS. 2A,3A OR 2B,3B                                         GROUP
      EPWK= EPWK + ESTPWI                                               GROUP
      EPW2K= EPW2K + ESTPWI**2                                          GROUP
      M1K= M1K + 1.0                                                    GROUP
  490 CONTINUE                                                          GROUP
```

```
C                                                                 GROUP
      IF ( M1K .EQ. 0.0 )  GO TO 900                              GROUP
C        EQS. 4A-6A  OR  4B-6B                                    GROUP
      SMPKPI= HISTPW*EPWK                                         GROUP
      SUMPK2= M1K*HISTPW**2                                       GROUP
      SUMPK = M1K*HISTPW                                          GROUP
  900 RETURN                                                      GROUP
      END                                                         GROUP
```

@ FOR IBETAI,1,760427, 39105 , 1

```
      SUBROUTINE IBETAI(X,A,B,P,IER)
C
C  COMPUTE INCOMPLETE BETA INTEGRAL FOR ARGUMENTS
C  X BETWEEN ZERO AND ONE, A AND B POSITIVE.
C
C  X  VALUE TO WHICH FUNCTION IS TO BE INTEGRATED
C  A  FIRST INPUT PARAMETER
C  B  SECOND INPUT PARAMETER
C  P  OUTPUT PROBABILITY THAT A RANDOM VARIABLE FROM A
C          BETA DISTRIBUTION HAVING PARAMETERS A AND B
C            WILL BE LESS THAN OR EQUAL TO X
C  IER  ERROR FLAG WITH POSSIBLE VALUES
C       =40 X NOT BETWEEN RANGE 0 TO 1
C       =50  A AND/OR B NOT POSITIVE
C       =60  GAMMA FUNCTION RANGE VIOLATED (NOT .GT. 0. AND .LE. 88.)
C
      LOGICAL INDEX
C  TEST FOR ADMISSIBILITY OF ARGUMENTS
      DATA ACU /1.E-8/
      P=X
      IER=50
      IF(A.LE.0.  .OR. B.LE.0.) RETURN
      IER=40
      IF(X.LE.0.  .OR. X.GE.1.) RETURN
      IER=0
C
C  CHANGE TAIL IF NECESSARY AND DETERMINE S
      ASB=A+B
      CX=1.0-X
      IF(A.GE.ASB*X) GO TO 10
      XX=CX
      CX=X
      AA=B
      BB=A
      INDEX=.TRUE.
      GO TO 20
   10 XX=X
      AA=A
      BB=B
      INDEX=.FALSE.
   20 TERM=1.0
      AI=1.0
      P=1.0
      NS=BB+CX*ASB
C
C  USE SOPER REDUCTION FORMULAE
      RX=XX/CX
   30 TEMP=BB-AI
      IF(NS.EQ.0) RX=XX
   40 TERM=TERM*TEMP*RX/(AA+AI)
      P=P+TERM
      TEMP=ABS(TERM)
      IF(TEMP.LE.ACU.AND.TEMP.LE.ACU*P) GO TO 50
      AI=AI+1.0
      NS=NS-1
      IF(NS.GE.0) GO TO 30
      TEMP=ASB
      ASB=ASB+1.0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          READ(SEGTRU)COUN4,IREG4,IZONE4                                    INITI
          IF(COUN4.NE.4HZZZZ) GO TO 10                                      INITI
          NARG=0                                                            INITI
          CALL ERRMES(4HCAMS,4HINIT,1,1)                                    INITI
          RETURN                                                            INITI
       10 CONTINUE                                                          INITI
          IF((IREG4.NE.STARTR.OR.IZONE4.NE.STARTZ).AND.STARTR.NE.0) GO TO 20INITI
          BACKSPACE SEGTRU                                                  INITI
    C                                                                       INITI
    C   PASS OVER HEADERS ON OTHER INPUT FILES                              INITI
    C   SAVE WINDOW NAMES FROM ACQUIS, OPEN DA FILE                         INITI
       25 CONTINUE                                                          INITI
          IF(IACQU.GT.0) GO TO 30                                           INITI
          REWIND ACQUIS                                                     INITI
          READ(ACQUIS)ISKP,ISKP,ICAS(2),ISKP,ISKP,HEAD                       INITI
          CALL TSAVE (0,0,IBAD)                                             INITI
       30 CONTINUE                                                          INITI
          IF(ICAMER.GT.0) GO TO 40                                          INITI
          REWIND CAMERR                                                     INITI
          READ(CAMERR)ISKP,ISKP,ICAS(3)                                     INITI
       40 CONTINUE                                                          INITI
          IF(ICROPW.GT.0) GO TO 50                                          INITI
          REWIND CROPW                                                      INITI
          READ(CROPW)ISKP,ISKP,ICAS(4)                                      INITI
       50 CONTINUE                                                          INITI
          IF(ISIG.GT.0) GO TO 60                                            INITI
          REWIND SIGEXT                                                     INITI
          READ(SIGEXT)ISKP,ISKP,ICAS(5)                                     INITI
       60 CONTINUE                                                          INITI
    C                                                                       INITI
    C   OUTPUT HEADER OF OUTPUT FILE                                        INITI
          REWIND CAMSF                                                      INITI
          ITOT=ICAMSF-15                                                    INITI
          WRITE(CAMSF)NAME(1),NAME(2),ICASE,IMODEL,                         INITI
         1   ICAS(3),ICAS(2),ICAS(4),ICAS(1),ICAS(5),IMULTI,ISCC,ICLASS,    INITI
         1  ISEXT,IACQ,ICAMS,(IFILL,I=1,ITOT)                               INITI
          RETURN                                                            INITI
          END                                                               INITI
```

@ FLT INIT,1,760427, 39023 , 1

```
      SUBROUTINE INIT                                                   INIT
C        INITIALIZATION ROUTINE.  INITIAIZES RANDOM NUMBER SEEDS        INIT
C        WRITES HEADER RECORDS, ETC.                                    INIT
C                                                                       INIT
C        COMMON BLOCK DEFINITIONS                                       INIT
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NIRIAL,RSTART,IPRINT,STARTP,STARTZ LEMCM
     2  ,ENDR   ,ENDZ   ,ISTG   ,ICAMS  ,IYFS   ,IACQ   ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECH,ICSFSH,ICSECE LEMCM
     4  ,ICSFYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C                                                                       INIT
C        LINKAGE ...   CALLED FROM LEM DRIVER                           INIT
C ********************************************************************* INIT
C                                                                       INIT
C                                                                       INIT
C        INITIALIZE RANDOM NUMBER SEEDS                                 INIT
      DO 10 I=1,7                                                       INIT
      SEED(I)= RSEED(I)                                                 INIT
   10 CONTINUE                                                          INIT
      RETURN                                                            INIT
      END                                                               INIT
```

```
@ FLT INITI,1,760427, 39047    , 1

      SUBROUTINE INITI(ISFG,IACQU,ICAMER,ICROPW,ISIG,HEAD,ITSFG)          INITI
C                                                                          INITI
C     THIS SUBROUTINE INITIALIZES THE INPUT FILES AND OUTPUT FILES,       INITI
C         READY TO PROCESS (READ PAST HEADERS,ETC.)                        INITI
C                                                                          INITI
C         CAMS CONTROL CARD INPUT DATA                                     CAMSCM
      COMMON/CAMSCM/ IMODFL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,         CAMSCM
     1    IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                       CAMSCM
      REAL MS                                                              CAMSCM
C                                                                          CAMSCM
C         FILE DEFINITIONS AND RECORD LENGTHS                              FILES
      COMMON  /FILES /                                                     FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ             FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO            FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD            FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                         FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT    FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                   FILES
C                                                                          FILES
C         ARGUMENT LIST FOR ERROR PROCESSING                               ARGLST
      COMMON  /ARGLST/                                                     ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                               ARGLST
      DIMENSION  IARG(10)                                                  ARGLST
      EQUIVALENCE  ( IARG,ARG )                                            ARGLST
C                                                                          ARGLST
      COMMON/SEGTRU/COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,                 SEGTRU
     1   TI,IPRIOR(6),ISPW,PI(2)                                           SEGTRU
C         LEN CONTROL CARD INPUT DATA                                      LENCM
      COMMON  /LENCM /                                                     LENCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ   LENCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC     LENCM
     3  ,ICASP ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE    LENCM
     4  ,ICSEYN,ICSESE,ICSLAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6    LENCM
     5  ,RSEED7,ICSEST,ICSLCO,ICSEYS,ICSECU,ICSECD                          LENCM
      DIMENSION  RSEED(7)                                                  LENCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5          LENCM
     1  ,RSEED6,RSEED7                                                     LENCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                        LENCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                            LENCM
C                                                                          LENCM
      COMMON/INDX/ INDEX(    1),IPOINT(2001),IPNT2(2001),IPEND,IPIN        INDX
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,               TRAINS
     1    ITWIN(4,25),ITTOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),           TRAINS
     1    TPTRUE,TTZULU(4),TPEST(4),TPERR(4),IERTOT(3),TM(3),TV(3),TB(3)  TRAINS
      INTEGER TTZULU                                                       TRAINS
      DIMENSION ITRAIN(129)                                                TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                            TRAINS
      COMMON/TACQ/IHOLD(4,129),IHP                                         TACQ
      DIMENSION HEAD( 4,4), NAME(2),ICAS(5)                                INITI
      DATA NAME/4HCAMS,4H OUT/                                             INITI
      DATA ICAS/5*0/                                                       INITI
      DATA IFILL/0/                                                        INITI
C                                                                          INITI
C    READ TO FIRST RECORD ON SEGTRU AND BACKSPACE                          INITI
      IF(ISFG.GT.0) GO TO 25                                               INITI
      REWIND SEGTRU                                                        INITI
      READ(SEGTRU)ISKP,ISKP,ICAS(1),ITSFG                                  INITI
   20 CONTINUE                                                             INITI
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      READ(SEGTRU)COUN4,IREG4,IZONE4
      IF(COUN4.NE.4HZZZZ) GO TO 10
      NARG=0
      CALL FRRMES(4HCAMS,4HINIT,1,1)
      RETURN
   10 CONTINUE
      IF((IREG4.NE.STARTR.OR.IZONE4.NE.STARTZ).AND.STARTR.NE.0) GO TO 20
      BACKSPACE SEGTRU
C
C  PASS OVER HEADERS ON OTHER INPUT FILES
C  SAVE WINDOW NAMES FROM ACQUIS, OPEN DA FILE
   25 CONTINUE
      IF(IACQU.GT.0) GO TO 30
      REWIND ACQUIS
      READ(ACQUIS)ISKP,ISKP,ICAS(2),ISKP,ISKP,HEAD
      CALL TSAVE (0,0,IBAD)
   30 CONTINUE
      IF(ICAMER.GT.0) GO TO 40
      REWIND CAMERR
      READ(CAMERR)ISKP,ISKP,ICAS(3)
   40 CONTINUE
      IF(ICROPW.GT.0) GO TO 50
      REWIND CROPW
      READ(CROPW)ISKP,ISKP,ICAS(4)
   50 CONTINUE
      IF(ISIG.GT.0) GO TO 60
      REWIND SIGEXT
      READ(SIGEXT)ISKP,ISKP,ICAS(5)
   60 CONTINUE
C
C  OUTPUT HEADER OF OUTPUT FILE
      REWIND CAMSF
      ITOT=LCAMSF-15
      WRITE(CAMSF)NAMF(1),NAMF(2),ICASE,IMODEL,
     1   ICAS(3),ICAS(2),ICAS(4),ICAS(1),ICAS(5),IMULT1,ISCC,ICLASS,
     1  ISEXT,IACQ,ICAMS,(IFILL,I=1,ITOT)
      RETURN
      END
```

```
      SUBROUTINE INPCHK                                                 INPCHK
C        SUBROUTINE INPCHK CHECKS THE VALIDITY OF THE INPUT PARAMETERS  INPCHK
C        ON THE LEM CONTROL CARDS. IT ALSO READS THE HEADER RECORDS OF  INPCHK
C        EACH REQUIRED INPUT FILE AND CHECKS THE FILENAME, CASE NUMBER, INPCHK
C        AND COUNTRY ON THAT FILE. ALSO INPCHK PRINTS THE INITIAL       INPCHK
C        PROBLEM STATUS INFORMATION                                     INPCHK
C                                                                       INPCHK
C        COMMON BLOCK DEFINITIONS                                       INPCHK
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C        CONSTANT QUANTITIES FOR LEM PROGRAM                            CONST
      COMMON  /CONST /                                                  CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CKOPW ,LCKOPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CKOPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                    IXCASF
      COMMON  /IXCASF/                                                  IXCASF
     1   IXCASF(  1)  ,LIXCAS                                           IXCASF
C                                                                       IXCASF
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)    ,ICASF ,CONTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECH,ICSESH,ICSECE LEMCM
     4  ,ICSEYH,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
```

```
     1   ITFR  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR                STATS
      EQUIVALENCE  ( NT,ITER )                                            STATS
C                                                                         STATS
C                                                                         INPCHK
C        INPUTS ...                                                       INPCHK
C           ALL OF THE QUANTITIES IN THE COMMON BLOCKS  /LEMCM/ AND       INPCHK
C           /FILES/  ARE REQUIRED BY INPCHK.                              INPCHK
C           IN ADDITION THE QUANTITIES  NTRMX, MAXR, AND MAXZ FROM        INPCHK
C           /CONST/  ARE REQUIRED.                                        INPCHK
C           ALSO DEPENDING UPON INPUT OPTIONS SELECTED, SOME OF THE       INPCHK
C           FOLLOWING FILES MAY BE INPUT TO INPCHK SO THEIR HEADER        INPCHK
C           RECORDS MAY BE CHECKED ..                                     INPCHK
C              SEGID , CROPW , SUBHST, CAMERR, YESERR, SIGEXT, ACQUIS     INPCHK
C              SEGTRU, CAMSF , YESOUT, CASF   , CASDIS                    INPCHK
C                                                                         INPCHK
C        OUTPUTS ...                                                      INPCHK
C          NERRS  = NO. OF NONFATAL ERRORS DETECTED ON LEM CONTROL CARDSINPCHK
C          NFATAL = NO. OF FATAL ERRORS DETECTED ON LEM CONTROL CARDS.E,INPCHK
C                                                                         INPCHK
C        LOCAL VARIABLE DEFINITIONS                                       INPCHK
C     FILL  = FILLER FOR HEADER RECORDS OF DATA FILES                     INPCHK
C     NFILL = NUMBER OF WORDS OF FILLER NECESSARY TO COMPLETE RECORD      INPCHK
C                                                                         INPCHK
C        LINKAGE ...   CALL INPCHK                                        INPCHK
C        INPCHK IS CALLED FROM INPUT                                      INPCHK
C                                                                         INPCHK
C        SUBROUTINES USED ...    ERRMES                                   INPCHK
C                                                                         INPCHK
C ******************************************************************** INPCHK
C                                                                         INPCHK
C                                                                         INPCHK
C        TEMPORARILY SET ICSECU AND ICSECD = ICASE                        INPCHK
      ICSECU= ICASE                                                       INPCHK
      ICSECD= ICASE                                                       INPCHK
C                                                                         INPCHK
      IF ( NTRIAL - RSTART .GT. NTRMX )                                   INPCHK
     1                               CALL ERRMES (3HLEM,6HINPCHK, 1,1)    INPCHK
C                                                                         INPCHK
      IF ( RSTART .GE. NTRIAL )                                           INPCHK
     1                               CALL ERRMES (3HLEM,6HINPCHK, 2,1)    INPCHK
C                                                                         INPCHK
      IF ( STARTR .LT. 0 )  GO TO 10                                      INPCHK
      IF ( ENDR .EQ. 0 )  GO TO 20                                        INPCHK
      IF ( STARTR .LE. ENDR .AND. ENDR .LE. MAXR )  GO TO 20              INPCHK
   10 CALL ERRMES (3HLEM,6HINPCHK, 3,1)                                   INPCHK
C                                                                         INPCHK
   20 IF ( STARTZ .LT. 0 )  GO TO 30                                      INPCHK
      IF ( ENDZ .EQ. 0 ) GO TO 40                                         INPCHK
      IF ( STARTZ .LE. ENDZ .AND. ENDZ .LE. MAXZ )  GO TO 40              INPCHK
   30 CALL ERRMES (3HLEM,6HINPCHK, 4,1)                                   INPCHK
C                                                                         INPCHK
   40 IF ( ISTG .LT. 0 .OR. ISTG .GT. 3 )  GO TO 50                       INPCHK
      IF ( ICAMS .LT. 0 .OR. ICAMS .GT. 3 )  GO TO 50                     INPCHK
      IF ( IYES .LT. 0 .OR. IYES .GT. 3 )  GO TO 50                       INPCHK
      GO TO 60                                                            INPCHK
C        ISTG, ICAMS, AND/OR IYES IS NOT 0, 1, 2, OR 3                    INPCHK
   50                                CALL ERRMES (3HLEM,6HINPCHK, 5,1)    INPCHK
C                                                                         INPCHK
   60 IF ( ICAMS .NE. 0 .AND. ISTG .EQ. 0 )                               INPCHK
     1                               CALL ERRMES (3HLEM,6HINPCHK, 6,1)    INPCHK
```

```
      C                                                                  INPCHK
   70 IF ( ICAMS .EQ. 2 )  GO TO 120                                     INPCHK
      IF ( ISTG .EQ. 2 )  GO TO 80                                       INPCHK
      IF ( ISTG .EQ. 1 .AND. RSTART .EQ. 0 )  GO TO 75                   INPCHK
      IF(ISTG .EQ. 3 .AND. RSTART .EQ. 0)GO TO 75                        INPCHK
      IF ( ISTG .NE. 0 )  GO TO 80                                       INPCHK
      C    ISTG= 0  OR  ISTG= 1 AND RSTART= 0.  READ AND CHECK HEADER    INPCHK
      C    OF THE SEGMENT ID FILE.                                       INPCHK
   75 NFILL= LSEGID - 4                                                  INPCHK
      REWIND SEGID                                                       INPCHK
      READ (SEGID)  ARG(3), ARG(4), IARG(1),ITSEG,  ( FILL,I=1,NFILL )   INPCHK
      ARG(6)= 5HSEGID                                                    INPCHK
      IF ( ARG(3) .NE. 4HSEGM .OR. ARG(4) .NE. 3HENT )                   INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1)  INPCHK
      NFILL= LSEGID - 1                                                  INPCHK
      READ (SEGID)   ARG(2), ( FILL,I=1,NFILL )                          INPCHK
      IF ( IARG(1) .NE. ICSESG .OR. ARG(2) .NE. CUNTRY )                 INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK, 7,1)  INPCHK
      REWIND SEGID                                                       INPCHK
      C                                                                  INPCHK
   80 IF ( ICAMS .EQ. 0 )  GO TO 85                                      INPCHK
      IF ( RSTART .GT. 0 )  GO TO 120                                    INPCHK
      C    ICAMS= 0  OR  ICAMS= 1 OR 3 AND RSTART = 0.  READ AND CHECK   INPCHK
      C    HEADER OF THE CROP WINDOW FILE.                               INPCHK
   85 NFILL= LCROPW - 3                                                  INPCHK
      REWIND CROPW                                                       INPCHK
      READ (CROPW)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )            INPCHK
      ARG(6)= 5HCROPW                                                    INPCHK
      IF ( ARG(3) .NE. 4HCROP .OR. ARG(4) .NE. 4HWIND )                  INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1)  INPCHK
      NFILL= LCROPW - 1                                                  INPCHK
      READ (CROPW)  ARG(2), ( FILL,I=1,NFILL )                           INPCHK
      IF ( IARG(1) .NE. ICSECW .OR. ARG(2) .NE. CUNTRY )                 INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK, 8,1)  INPCHK
      REWIND CROPW                                                       INPCHK
      C                                                                  INPCHK
      C    READ AND CHECK HEADER OF THE CAMS ERROR MODEL FILE.           INPCHK
   90 NFILL= LCAMER - 3                                                  INPCHK
      REWIND CAMERR                                                      INPCHK
      READ (CAMERR)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )           INPCHK
      ARG(6)= 6HCAMERR                                                   INPCHK
      IF ( ARG(3) .NE. 4HCAMS .OR. ARG(4) .NE. 4H ERR )                  INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1)  INPCHK
      NFILL= LCAMER - 1                                                  INPCHK
      READ (CAMERR)  ARG(2), ( FILL,I=1,NFILL )                          INPCHK
      IF ( IARG(1) .NE. ICSECE .OR. ARG(2) .NE. CUNTRY )                 INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK, 9,1)  INPCHK
      REWIND CAMERR                                                      INPCHK
      C                                                                  INPCHK
      C    READ AND CHECK HEADER OF THE SIGNATURE EXTENSION FILE         INPCHK
  100 NFILL= LSIGEX - 3                                                  INPCHK
      REWIND SIGEXT                                                      INPCHK
      READ (SIGEXT)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )           INPCHK
      ARG(6)= 6HSIGEXT                                                   INPCHK
      IF ( ARG(3) .NE. 4HSIGE .OR. ARG(4) .NE. 4HXTEN )                  INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1)  INPCHK
      NFILL= LSIGEX - 1                                                  INPCHK
      READ (SIGEXT)  ARG(2), ( FILL,I=1,NFILL )                          INPCHK
      IF ( IARG(1) .NE. ICSESF .OR. ARG(2) .NE. CUNTRY )                 INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,10,1)  INPCHK
```

```
      REWIND SIGEXT                                                     INPCHK
C                                                                       INPCHK
C        READ AND CHECK HEADER OF THE DATA ACQUISITION FILE             INPCHK
  110 NFILL= LACQ - 3                                                   INPCHK
      REWIND ACQUIS                                                     INPCHK
      READ (ACQUIS)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )          INPCHK
      ARG(6)= 6HACQUIS                                                  INPCHK
      IF ( ARG(3) .NE. 4HACQU .OR. ARG(4) .NE. 3HISI )                  INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,19,1)   INPCHK
      NFILL= LACQ - 1                                                   INPCHK
      READ (ACQUIS)  ARG(2), ( FILL,I=1,NFILL )                         INPCHK
      IF ( IARG(1) .NE. ICSEAC .OR. ARG(2) .NE. CUNTRY )                INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,11,1)   INPCHK
      REWIND ACQUIS                                                     INPCHK
C                                                                       INPCHK
  120 IF ( IYES .EQ. 0 )  GO TO 125                                     INPCHK
      IF ( IYES .EQ. 2 )  GO TO 130                                     INPCHK
      IF ( RSTART .GT. 0 )  GO TO 130                                   INPCHK
C        IYES= 0  OR  IYES= 1 OR 3 AND RSTART = 0.  READ AND CHECK      INPCHK
C        HEADER OF THE YES ERROR MODEL FILE.                            INPCHK
  125 NFILL= LYESER - 3                                                 INPCHK
      REWIND YESERR                                                     INPCHK
      READ (YESERR)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )          INPCHK
      ARG(6)= 6HYESERR                                                  INPCHK
      IF ( ARG(3) .NE. 4HYESE .OR. ARG(4) .NE. 4HRROR )                 INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,19,1)   INPCHK
      NFILL= LYESER - 1                                                 INPCHK
      READ (YESERR)  ARG(2), ( FILL,I=1,NFILL )                         INPCHK
      IF ( IARG(1) .NE. ICSEYM .OR. ARG(2) .NE. CUNTRY )                INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,12,1)   INPCHK
      REWIND YESERR                                                     INPCHK
C                                                                       INPCHK
C        READ AND CHECK HEADER OF SUBSTRATA HISTORICAL FILE             INPCHK
  130 NFILL= LSUBH - 3                                                  INPCHK
      REWIND SUBHST                                                     INPCHK
      READ (SUBHST)  ARG(3),ARG(4),IARG(1),IHXSEG                       INPCHK
      ARG(6)= 6HSUBHST                                                  INPCHK
      IF ( ARG(3) .NE. 4HSUB  .OR. ARG(4) .NE. 4HHIST )                 INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,19,1)   INPCHK
      NFILL= LSUBH - 1                                                  INPCHK
      READ (SUBHST)  ARG(2), ( FILL,I=1,NFILL )                         INPCHK
      IF (IARG(1) .NE. ICSESH .OR. ARG(2) .NE. CUNTRY )                 INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,13,1)   INPCHK
      REWIND SUBHST                                                     INPCHK
C                                                                       INPCHK
  140 IF ( ICAMS .EQ. 2 )  GO TO 155                                    INPCHK
      IF ( ISTG .EQ. 2 )  GO TO 145                                     INPCHK
      IF ( RSTART .EQ. 0 )  GO TO 150                                   INPCHK
      IF ( ISTG .EQ. 0 )  GO TO 150                                     INPCHK
C        ISTG = 2  OR ISTG = 1 OR 3 AND RSTART .GT. 0                   INPCHK
  145 NFILL= LSEGTR - 3                                                 INPCHK
      REWIND SEGTRU                                                     INPCHK
      READ (SEGTRU)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )          INPCHK
      ARG(6)= 6HSEGTRU                                                  INPCHK
      IF ( ARG(3) .NE. 4HSEGT .OR. ARG(4) .NE. 4HRUTH )                 INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,19,1)   INPCHK
      NFILL= LSEGTR - 1                                                 INPCHK
      READ (SEGTRU)  ARG(2), ( FILL,I=1,NFILL )                         INPCHK
      IF ( IARG(1) .NE. ICSEST .OR. ARG(2) .NE. CUNTRY )                INPCHK
     1                              CALL ERRMES (3HLEM,6HINPCHK,14,1)   INPCHK
```

```
      REWIND SEGTRU                                                     INPCHK
C                                                                       INPCHK
  150 IF ( RSTART .EQ. 0 )  GO TO 160                                   INPCHK
      IF ( ICAMS .EQ. 0 )  GO TO 160                                    INPCHK
C        ICAMS = 2  OR ICAMS = 1 OR 3 AND RSTART .GT. 0                 INPCHK
  155 NFILL= LCAMSF - 3                                                 INPCHK
      REWIND CAMSF                                                      INPCHK
      READ (CAMSF)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )           INPCHK
      ARG(6)= 5HCAMSF                                                   INPCHK
      IF ( ARG(3) .NE. 4HCAMS .OR. ARG(4) .NE. 4H OUT )                 INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1) INPCHK
      NFILL= LCAMSF - 1                                                 INPCHK
      READ (CAMSF)  ARG(2), ( FILL,I=1,NFILL )                          INPCHK
      IF ( IARG(1) .NE. ICSFCO .OR. ARG(2) .NE. CUNTRY )                INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,15,1) INPCHK
      REWIND CAMSF                                                      INPCHK
C                                                                       INPCHK
  160 IF ( IYES .EQ. 2 )  GO TO 165                                     INPCHK
      IF ( RSTART .EQ. 0 )  GO TO 170                                   INPCHK
      IF ( IYES .EQ. 0 )  GO TO 170                                     INPCHK
C        IYES = 2  OR IYES = 1 OR 3 AND RSTART .GT. 0                   INPCHK
  165 NFILL= LYESO - 3                                                  INPCHK
      REWIND YESOUT                                                     INPCHK
      READ (YESOUT)  ARG(3),ARG(4),IARG(1), ( FILL,I=1,NFILL )          INPCHK
      ARG(6)= 6HYESOUT                                                  INPCHK
      IF ( ARG(3) .NE. 3HYES .OR. ARG(4) .NE. 1H  )                     INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,19,1) INPCHK
      NFILL= LYESO - 1                                                  INPCHK
      READ (YESOUT)  ARG(2), ( FILL,I=1,NFILL )                         INPCHK
      IF ( IARG(1) .NE. ICSEYS .OR. ARG(2) .NE. CUNTRY )                INPCHK
     1                                CALL ERRMES (3HLEM,6HINPCHK,16,1) INPCHK
      REWIND YESOUT                                                     INPCHK
C                                                                       INPCHK
C        OPEN CAS CUMULATIVE FILE                                       INPCHK
  170 DEFINE FILE 14(388,504,U,IDUM)                                           *NEW
C                                                                       INPCHK **-1
      IF ( RSTART .EQ. 0 )  GO TO 900                                   INPCHK
C                                                                       INPCHK
C        READ AND CHECK HEADER RECORD OF CAS CUMULATIVE FILE            INPCHK
      CALL RANACF (CASF,1,CASCUM,LCASF,IXCASF,LIXCAS,1)                 INPCHK
      COUN= CASCUM(3)                                                   INPCHK
      NT= ICASC(4)                                                      INPCHK
      ARG(6)= 4HCASF                                                    INPCHK
      ARG(3)= CASCUM(1)                                                 INPCHK
C        SHIFT FILENAME 4 CHARACTERS (24 BITS) TO THE LEFT              INPCHK
C        BY MULTIPLYING BY 2**24                                        INPCHK
      IARG(4)= ICASC(1)*16777216                                        INPCHK
      IF ( CASCUM(1) .NE. 6HCASCUM )  CALL ERRMES (3HLEM,6HINPCHK,19,1) INPCHK
      IARG(1)= ICASC(2)                                                 INPCHK
      IARG(2)= COUN                                                     INPCHK
      IF ( ICASC(2) .NE. ICSECU .OR. COUN .NE. CUNTRY )                 INPCHK
     1    CALL ERRMES (3HLEM,6HINPCHK,17,1)                             INPCHK
      IF ( NT .NE. RSTART )  CALL ERRMES (3HLEM,6HINPCHK,21,1)          INPCHK
C                                                                       INPCHK
  900 RETURN                                                            INPCHK
      END                                                               INPCHK
```

```
      SUBROUTINE INPERR (ICODE)                                         INPERR
C        PRINTS ERROR MESSAGES FOR THE LEM INPUT PROCESSOR              INPERR
C                                                                       INPERR
C        INPUT PARAMETER ...                                            INPERR
C        ICODE = ERROR MESSAGE CODE                                     INPERR
C                                                                       INPERR
C        COMMON BLOCK DEFINITIONS                                       INPERR
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        CONSTANT QUANTITIES FOR LEM PROGRAM                            CONST
      COMMON  /CONST /                                                  CONST
     1   NTRAX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO          FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACU  ,LTACU ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACU  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)    ,ICASF ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISIG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYLS,IPRCAS,ICSESG,ICSECR,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NI,ITER )                                          STATS
C                                                                       STATS
C                                                                       INPERR
C        LOCAL VARIABLES                                                INPERR
C     IMES  = ERROR MESSAGE CODE                                        INPERR
C     MXLEM = MAXIMUM ERROR MESSAGE CODE                                INPERR
C                                                                       INPERR
      DATA  MXLEM  / 21 /                                               INPERR
C                                                                       INPERR
C        LINKAGE ...   CALL INPERR (ICODE)                              INPERR
C        INPERR IS CALLED FROM ERRMES, WHICH IS CALLED FROM INPCHK      INPERR
C                                                                       INPERR
C ******************************************************************** INPERR
C                                                                       INPERR
C                                                                       INPERR
C        MOST ERROR MESSAGES OCCUPY 2 OR 3 LINES.                       INPERR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IMES= ICODE                                                       INPERR
      IF ( IMES .LT. 0 .OR. IMES .GT. MXLEM  )  GO TO 980               INPERR
C       BRANCH TO PRINT PROPER MESSAGE                                  INPERR
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170 INPERR
     1  ,180,190,200,210                                                INPERR
     2  ), IMES                                                         INPERR
C                                                                       INPERR
 10   WRITE (OUTP,1001)  NTRIAL,RSTART,NTRMX                            INPERR
 1001 FORMAT (48H0TOO MANY MONTE CARLO TRIALS REQUESTED.  NTRIAL=I4,    INPERR
     1    9H  RSTART=I4/30H MAX. NO. OF TRIALS PER RUN IS,I4)           INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 20   WRITE (OUTP,1002)  RSTART,NTRIAL                                  INPERR
 1002 FORMAT (8H0RSTART=I4,27H  MUST BE LESS THAN NTRIAL=I4)            INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 30   WRITE (OUTP,1003)  STARTR,ENDR,MAXR                               INPERR
 1003 FORMAT (8H0STARTR=I4,29H  MUST BE BETWEEN 0 AND ENDR=I4,          INPERR
     1  20H.  ENDR MUST BE .LE.,I4)                                     INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 40   WRITE (OUTP,1004)  STARTZ,ENDZ,MAXZ                               INPERR
 1004 FORMAT (8H0STARTZ=I4,29H  MUST BE BETWEEN 0 AND ENDZ=I4,          INPERR
     1  20H.  ENDZ MUST BE .LE.,I4)                                     INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 50   WRITE (OUTP,1005)  ISTG,ICAMS,IYES                                INPERR
 1005 FORMAT (6H0ISTG=I2,9H,  ICAMS=I2,12H,  AND IYES=I2,26H  MUST ALL BINPERR
     1E 0,1,2, OR 3.)                                                   INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 60   CALL PAGER (1)                                                    INPERR
      WRITE (OUTP,1006)  ICAMS,ISTG                                     INPERR
 1006 FORMAT (48H0IF ICAMS IS NONZERO, THEN ISTG MUST BE NONZERO./      INPERR
     1  81H I.E. IF THE CAMS ERRORS ARE HELD CONSTANT, THEN SO MUST THE INPERR
     2SEGMENT TRUTH ERROR./8H  ICAMS=I5,8H   ISTG=I5)                   INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 70   WRITE (OUTP,1007)  IARG(1),IARG(2),ICSESG,CUNTRY                  INPERR
 1007 FORMAT (13H0CASE NUMBER=I5,13H  OR COUNTRY ,A6,57H  FROM SEGMENT IINPERR
     1D FILE DOES NOT AGREE WITH INPUTS ICSESG=I5/14H  AND CUNTRY= A6)  INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 80   WRITE (OUTP,1008)  IARG(1),IARG(2),ICSECW,CUNTRY                  INPERR
 1008 FORMAT (13H0CASE NUMBER=I5,13H  OR COUNTRY ,A6,58H  FROM CROP WINDINPERR
     1OW FILE DOES NOT AGREE WITH INPUTS ICSECW=I5/14H  AND CUNTRY= A6) INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 90   WRITE (OUTP,1009)  IARG(1),IARG(2),ICSECE,CUNTRY                  INPERR
 1009 FORMAT (13H0CASE NUMBER=I5,13H  OR COUNTRY ,A6,57H  FROM CAMS ERROINPERR
     1R FILE DOES NOT AGREE WITH INPUTS ICSECE=I5/14H  AND CUNTRY= A6)  INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 100  WRITE (OUTP,1010)  IARG(1),IARG(2),ICSESE,CUNTRY                  INPERR
 1010 FORMAT (13H0CASE NUMBER=I5,13H  OR COUNTRY ,A6,62H  FROM THE SIGNAINPERR
     1TURE EXTENSION FILE DOES NOT AGREE WITH INPUTS /                  INPERR
     2   9H  ICSESE=I5,14H  AND CUNTRY= A6)                             INPERR
      GO TO 999                                                         INPERR
C                                                                       INPERR
 110  WRITE (OUTP,1011)  IARG(1),IARG(2),ICSLAC,CUNTRY                  INPERR
```

```
      1011 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,67H  FROM THE DATA INPERR
          1ACQUISITION FILE DOES NOT AGREE WITH INPUTS ICSEAC=I5/             INPERR
          2  14H  AND CUNTRY= A6)                                             INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       120  WRITE (OUTP,1020)  IARG(1),IARG(2),ICSEYM,CUNTRY                  INPERR
      1020 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,66H  FROM THE YES EINPERR
          1RROR MODEL FILE DOES NOT AGREE WITH INPUTS ICSEYM=I5/              INPERR
          2  14H  AND CUNTRY= A6)                                             INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       130  WRITE (OUTP,1030)  IARG(1),IARG(2),ICSESH,CUNTRY                  INPERR
      1030 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,63H  FROM THE SUBSTINPERR
          1RATA HISTORICAL FILE DOES NOT AGREE WITH INPUTS/                   INPERR
          2  9H  ICSESH=I5,14H  AND CUNTRY= A6)                               INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       140  WRITE (OUTP,1040)  IARG(1),IARG(2),ICSEST,CUNTRY                  INPERR
      1040 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,64H  FROM THE SEGMEINPERR
          1NT TRUTH FILE DOES NOT AGREE WITH INPUTS ICSEST=I5/14H  AND CUNTRYINPERR
          1= A6)                                                              INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       150  WRITE (OUTP,1050)  IARG(1),IARG(2),ICSECO,CUNTRY                  INPERR
      1050 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,62H  FROM THE CAMS INPERR
          1OUTPUT FILE DOES NOT AGREE WITH INPUTS ICSECO=I5/14H  AND CUNTRY= INPERR
          2 A6)                                                               INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       160  WRITE (OUTP,1060)  IARG(1),IARG(2),ICSEYS,CUNTRY                  INPERR
      1060 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,61H  FROM THE YES OINPERR
          1UTPUT FILE DOES NOT AGREE WITH INPUTS ICSEYS= I5/14H  AND CUNTRY =INPERR
          2 A6)                                                               INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       170  WRITE (OUTP,1070)  IARG(1),IARG(2),ICSEYS,CUNTRY                  INPERR
      1070 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,65H  FROM THE CAS CINPERR
          1UM OUTPUT FILE DOES NOT AGREE WITH INPUTS ICASE = I5/              INPERR
          2  14H  AND CUNTRY= A6)                                             INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       180  WRITE (OUTP,1080)  IARG(1),IARG(2),ICSECD,CUNTRY                  INPERR
      1080 FORMAT (13HOCASE NUMBER=I5,13H  OR COUNTRY ,A6,66H  FROM THE CAS DINPERR
          1IST OUTPUT FILE DOES NOT AGREE WITH INPUTS ICASE = I5/             INPERR
          2  14H  AND CUNTRY= A6)                                             INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
     C                                                                        INPERR
       190  WRITE (OUTP,1190)  ARG(6),ARG(3),ARG(4)                           INPERR
      1190 FORMAT (31HOIMPROPER HEADER LABEL ON FILE ,A6,10H,  LABEL =2A4)    INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       200  WRITE (OUTP,1200)  ARG(1)                                         INPERR
      1200 FORMAT (81HOIMPROPER LABEL AND SEQUENCE NUMBER ON A LEM CONTROL CAINPERR
          1RD,  LABEL AND SEQ. NO. = A6)                                      INPERR
           GO TO 999                                                          INPERR
     C                                                                        INPERR
       210  WRITE (OUTP,1210)  ARG(3),NT,RSTART                               INPERR
      1210 FORMAT (26HOITERATION NUMBER NT FROM ,A6,7H FILE = I4,             INPERR
          1  30H  DOES NOT AGREE WITH RSTART = I4,23H  FROM LEM CONTROL CARD)INPERR
```

```
          GO TO 999                                                            INPERR
C                                                                              INPERR
  980     WRITE (OUTP,1980)  IMES                                              INPERR
 1980     FORMAT(52H0 ERROR IN SUBR. INPERR.  ILLEGAL ERROR MESSAGE CODE,I5)INPERR
C                                                                              INPERR
  999     RETURN                                                               INPERR
          END                                                                  INPERR
```

```
      SUBROUTINE INPT(ISEG,IACQU,ICAMER,ICROPW,ISIGX,IPASS,                 INP
     1  IDONE,IEND)                                                         INP
C                                                                           INP
C   THIS SUBROUTINE GETS THE NEXT SET OF RECORDS TO PROCESS FROM            INP
C     THE INPUT FILES.                                                      INP
C                                                                           INP
      COMMON/SEGTRU/COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,                  SEGTRU
     1  IT,IPRIOR(6),ISPW,PT(2)                                             SEGTRU
      COMMON/ACQUIS/COUN1,IREG1,IZONE1,ISTRA1,ISUB1,ISEG1,                  ACQUIS
     1    IWIN(4,25),ITOTAL                                                 ACQUIS
      COMMON/CAMERR/ COUN2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,                 CAMERR
     1  PW(3,4),BERR(3,4),SIGERR(3,4)                                       CAMERR
      COMMON/CROPW/COUN3,IREG3,IZONE3,ISTRA3,ISUB3,                         CROPW
     1  START(2,4),END(2,4),SD(2),ERR(2,5)                                  CROPW
      INTEGER START,END,SD,ERR                                              CROPW
      COMMON/SIGEX/COUN5,IREG5,IZONE5,ZB(3,2),ZSIG(3,2,6)                   SIGEX
C       FILE DEFINITIONS AND RECORD LENGTHS                                 FILES
      COMMON  /FILES /                                                      FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ              FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASE  ,LCASE ,YESOUT,LYESO             FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD             FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                          FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASE  ,YESOUT      FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                    FILES
C                                                                           FILES
      COMMON/INDX/ INDEX(    1),IPOINT(2001),IPNT2(2001),IPEND,IPIN         INDX
      COMMON/TACQ/IHOLD(4,129),IHP                                          TACQ
C       LEM CONTROL CARD INPUT DATA                                         LEMCM
      COMMON  /LEMCM /                                                      LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ   ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3  ,IPRCAM,IPRYES,IPRCAS,ICSESC,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSFCD                          LEMCM
      DIMENSION  RSEED(7)                                                   LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5 ,         LEMCM
     1  ,RSEED6,RSEED7                                                      LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                         LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                             LEMCM
C                                                                           LEMCM
C       CAMS CONTROL CARD INPUT DATA                                        CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,           CAMSCM
     1     IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                       CAMSCM
      REAL MS                                                               CAMSCM
C                                                                           CAMSCM
C       ARGUMENT LIST FOR ERROR PROCESSING                                  ARGLST
      COMMON  /ARGLST/                                                      ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                                ARGLST
      DIMENSION  IARG(10)                                                   ARGLST
      EQUIVALENCE  ( IARG,ARG )                                             ARGLST
C                                                                           ARGLST
C       STATISTICAL INFORMATION FOR LEM                                     STATS
      COMMON  /STATS /                                                      STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR                  STATS
      EQUIVALENCE  ( NI,ITER )                                              STATS
C                                                                           STATS
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,                TRAINS
     1    ITWIN(4,25),ITTOT,TMM(3,4,25),TBP(3,4,25),TVV(3,4,25),            TRAINS
```

```
     1   TPTRUE,TIZULU(4),TPEST(4),TPERR(4),TERTOT(3),TH(3),TV(3),IB(3) TRAINS
      INTEGER TIZULU                                                 TRAINS
      DIMENSION ITRAIN(129)                                          TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                      TRAINS
   30 CONTINUE                                                       INP
      IF(ISEG.GT.0) GO TO 10                                         INP
      IF(IREG4.EQ.ENDR.AND.IZONE4.EQ.ENDZ) IEND=1                    INP
      READ(SEGTRU)COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,             INP
     1  IT,(IPRIOR(I),I=1,6),ISPH,(PT(I),I=1,2)                      INP
C                                                                    INP
C  CHECK IF DONE                                                     INP
      IF(ENDZ.EQ.0.AND.COUN4.NE.4HZZZZ) GO TO 10                     INP
      IF(ENDZ.EQ.0.AND.COUN4.EQ.4HZZZZ) GO TO 20                     INP
      IF(IEND.EQ.0.AND.COUN4.EQ.4HZZZZ) GO TO 21                     INP
      IF((IEND.EQ.1).AND.(IREG4.NE.ENDR.OR.IZONE4.NE.ENDZ)           INP
     1  .AND.COUN4.NE.4HZZZZ) GO TO 21                               INP
      IF (COUN4.EQ.4HZZZZ) GO TO 20                                  INP
      GO TO 10                                                       INP
C                                                                    INP
C  ERROR RETURN - NO END ZONE                                        INP
   21 CONTINUE                                                       INP
      IDONE=2                                                        INP
      NARG=0                                                         INP
      CALL FRRMES(4HCAMS,4HINPT,2,0)                                 INP
C                                                                    INP
C  DONE PROCESSING                                                   INP
   20 CONTINUE                                                       INP
      IDONE=1                                                        INP
      RETURN                                                         INP
C                                                                    INP
C  CHECK IF TRAINING OR ORDINARY SEGMENT, AND BRANCH                 INP
   10 CONTINUE                                                       INP
C                                                                    INP
C  ACQUIS. PASS OR SPECIAL CASE - ORDINARY SEGMENT                   INP
      IF(IPASS.EQ.0) GO TO 200                                       INP
C                                                                    INP
C  TRAINING PASS - SKIP OVER ORDINARY'S                              INP
      IF(IPASS.EQ.1.AND.IT.EQ.1) GO TO 30                            INP
      IF(IPASS.EQ.1.AND.IT.EQ.0) GO TO 200                           INP
C                                                                    INP
C  ORDINARY PASS                                                     INP
      IF(IPASS.EQ.2.AND.IT.EQ.1) GO TO 200                           INP
C                                                                    INP
C  ORDINARY PASS - TRAINING SEGMENTS - WRITE TO OUTPUT FILE          INP
C                        GET FROM SCRATCH FILE                       INP
      CALL TSAVE(ISEG4,1,IBAD)                                       INP
      WRITE(CAMSF)COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,             INP
     1  TPTRUE,(TIZULU(I),TPEST(I),TPERR(I),I=1,4)                   INP
      NCAMSR=NCAMSR+1                                                INP
      GO TO 30                                                       INP
  200 CONTINUE                                                       INP
C                                                                    INP
C  READ ACQUIS. FILE                                                 INP
      IF(IACQU.GT.0) GO TO 400                                       INP
      READ(ACQUIS)COUN1,IREG1,IZONE1,ISTRA1,ISUB1,ISEG1,             INP
     1   ((IWIN(I,J),J=1,25),I=1,4),ITOTAL                           INP
C                                                                    INP
C  MISSING RECORD(                                                   INP
      IARG(1)=6HACQUIS                                               INP
      IF(COUN1.EQ.4HZZZZ) GO TO 310                                  INP
```

```
      IF(ISEG1.NE.ISEG4) GO TO 200
      IF(COUN1.EQ.4HZZZZ) GO TO 310
      GO TO 400
C
C   MISSING RECORD ERROR RETURN
  310 CONTINUE
      NARG=1
      IDONE=2
      CALL ERRMES(4HCAMS,4HINPT,3,1)
      RETURN
C
C   READ CAMERR FILE
  400 CONTINUE
      IF(IACQU.LE.0)NREC(7)=NREC(7)+1
      IF(ICAMER.GT.0) GO TO 500
      DO 430 I=1,4
      PW(1,I)=PT(I)/100.
      PW(2,I)=0.
      PW(3,I)=0.
  430 CONTINUE
      DO 432 I=2,3
      DO 434 J=1,4
      BERR(I,J)=0.
      SIGERR(I,J)=0.
  434 CONTINUE
  432 CONTINUE
  410 CONTINUE
      IF(IMODEL.EQ.2) GO TO 420
      READ(CAMERR)COUN2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,
     1   (((PW(I,J),I=1,3),(BERR(I,J),SIGERR(I,J),I=1,3),ISKP,
     1   ISKP),J=1,4)
      GO TO 440
  420 CONTINUE
      READ(CAMERR) COUN2,IREG2,IZONE2,ISTRA2,ISUB2,ISEG2,
     1   ((ISKP,I=1,9),BERR(1,J),SIGERR(1,J),J=1,4)
  440 CONTINUE
      IARG(1)=6HCAMERR
      IF(COUN2.EQ.4HZZZZ) GO TO 310
      IF(ISEG2.NE.ISEG4) GO TO 410
      NREC(4)=NREC(4)+1
C
C   READ CROPW FILE
  500 CONTINUE
      IF(ICROPW.GT.0) GO TO 600
      IF(ISUB3.EQ.ISUB4) GO TO 600
      READ(CROPW)COUN3,IREG3,IZONE3,ISTRA3,ISUB3,
     1  (((START(I,J),END(I,J),J=1,4),SD(I),(ERR(I,J),J=1,5)),I=1,2)
      IARG(1)=5HCROPW
      IF(COUN3.EQ.4HZZZZ) GO TO 310
      IF(ISUB3.NE.ISUB4) GO TO 500
      NREC(2)=NREC(2)+1
  600 CONTINUE
      IARG(1)=5HCROPW
C   CHECK IF NO DATA - WINTER/SPRING
      IF(ICROPW.LE.0.AND.START(1+ISPW,2).EQ.0.) GO TO 310
C
C   READ SIGEXT FILE
      IF(ISIGX.GT.0) GO TO 700
      IF(IZONE5.EQ.IZONE4) GO TO 700
      DO 642 I=2,3
```

```
      ZB(I,1)=-1.                                                       INP
      DO 644 J=1,2                                                      INP
      DO 646 K=1,6                                                      INP
      ZSIG(I,J,K)=0.                                                    INP
  646 CONTINUE                                                          INP
  644 CONTINUE                                                          INP
  642 CONTINUE                                                          INP
  640 CONTINUE                                                          INP
      IF(IMODEL.EQ.2) GO TO 610                                         INP
      READ(SIGEXT)COUNS,IREGS,IZONES,                                   INP
     1  ((ZB(I,J),J=1,2),((ZSIG(I,J,K),K=1,6),J=1,2),I=1,3)            INP
      GO TO 620                                                         INP
  610 CONTINUE                                                          INP
      READ(SIGEXT)COUNS,IREGS,IZONES,                                   INP
     1  (ISKP,I=1,42),(ZB(1,J),J=1,2),((ZSIG(1,J,K),K=1,6),J=1,2)      INP
  620 CONTINUE                                                          INP
       IARG(1)=5HSIGEX                                                  INP
      IF(COUNS.EQ.4HZZZZ) GO TO 310                                     INP
      IF(IZONES.NE.IZONE4) GO TO 640                                    INP
      DO 634 I=1,3                                                      INP
      ZB(I,1)=ZB(I,1)+1.                                                INP
  634 CONTINUE                                                          INP
      NREC(6)=NREC(6)+1                                                 INP
  700 CONTINUE                                                          INP
C                                                                       INP
C  CHECK RANGE ON PT(2)                                                 INP
      IDONE=0                                                           INP
      IF(PT(2)*PW(2,IWIND).LT.(PT(2)+PT(1)-100.))                       INP
     1  PT(2)=(100.-PT(1))/(100.-PW(2,IWIND))                           INP
      IF(PT(2)*PW(2,IWIND).GT.PT(1)) PT(2)=PT(1)/PW(2,IWIND)            INP
      RETURN                                                            INP
      END                                                               INP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INPUT                                                  INPUT
C        READS AND CHECKS LEM CONTROL CARD INPUT                        INPUT
C        ALSO CALL ROUTINES TO READ CONTROL CARD INPUT FOR CAMS AND CAS.INPUT
C                                                                       INPUT
C        COMMON BLOCK DEFINITIONS                                       INPUT
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)    ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISIG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSFSG,ICSECW,ICSFSW,ICSECE LEMCM
     4  ,ICSEYM,ICSFSF,ICSLAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C                                                                       INPUT
C        LOCAL VARIABLES                                                INPUT
C     LBL   = CONTROL CARD LABEL AND SEQUENCE NUMBER IN A6 FORMAT       INPUT
      REAL  LBL,LBL1                                                    INPUT
C                                                                       INPUT
C        LINKAGE ...  CALL INPUT                                        INPUT
C        INPUT IS CALLED FROM THE LEM DRIVER                            INPUT
C                                                                       INPUT
C        SUBROUTINES CALLED ...                                         INPUT
C        EJECT, ERRMES, INPCHK, CAMSIN, CASIN                           INPUT
C                                                                       INPUT
C *******************************************************************   INPUT
C                                                                       INPUT
C                                                                       INPUT
C        READ FIRST LEM CONTROL CARD                                    INPUT
      READ (INP,1)  TITLE,LBL1                                          INPUT
    1 FORMAT (10A6,14X,A6)                                              INPUT
C                                                                       INPUT
C        READ SECOND LEM CONTROL CARD                                   INPUT
      READ (INP,2)  ICASE,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ     INPUT
     1  ,ENDR,ENDZ,ISIG,ICAMS,IYES,IACQ,ICLASS,ISEXT,ISCC,ICAS2,ICAS3   INPUT
```

```
     2  ,IPRCAM,IPRYES,IPRCAS,LBL                                          INPUT
 2    FORMAT (I4,1X,A6,2I4,I3,4I4,12I3,A6)                                 INPUT
C                                                                          INPUT
C         CHECK LABEL AND SEQUENCE NUMBER OF FIRST CONTROL CARD.           INPUT
      IF ( LBL1 .EQ. 6HLEM 01 )  GO TO 200                                 INPUT
      ARG(1)= LBL1                                                         INPUT
      NARG= 1                                                              INPUT
      CALL ERRMES (3HLEM,5HINPUT,20,1)                                     INPUT
C                                                                          INPUT
200   NPAGE= 0                                                             INPUT
      CALL EJECT (17)                                                      INPUT
      WRITE (OUTP,10)                                                      INPUT
10    FORMAT (/ 20X,45HL E M   I N P U T   C O N T R O L   C A R D S/)     INPUT
      WRITE (OUTP,11)  TITLE,LBL1                                          INPUT
11    FORMAT (/7H  TITLE/2X,10A6,14X,A6)                                   INPUT
      WRITE (OUTP,12) ICASE,CONTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ      INPUT
     1  ,ENDR,END7,ISTG,ICAMS,IYES,IACQ,ICLASS,ISEXT,ISCC,ICAS2,ICAS3      INPUT
     2  ,IPRCAM,IPRYES,IPRCAS,LBL                                          INPUT
12    FORMAT(107H0ICASE  COUNTRY  NTRIAL  RSTART  IPRINT  STARTR  STARTZINPUT
     1  ENDR  END7  ISTG  ICAMS  IYES  IACQ  ICLASS  ISEXT /               INPUT
     2  I6,3X,A6,I7,I8,I7,I9,I8,I7,I6,I5,I6,I7,I6,I7,I7/                  INPUT
     3  52H  ISCC  ICAS2  ICAS3  IPRCAM  IPRYES  IPRCAS   LABEL/           INPUT
     4  I5,I6,I7,I8,I8,I8,4X,A6)                                           INPUT
C         CHECK LABEL AND SEQUENCE NUMBER OF SECOND CONTROL CARD           INPUT
      IF ( LBL .EQ. 6HLEM 02 )  GO TO 300                                  INPUT
      ARG(1)= LBL                                                          INPUT
      NARG= 1                                                              INPUT
      CALL ERRMES (3HLEM,5HINPUT,20,1)                                     INPUT
C                                                                          INPUT
C         READ THIRD CONTROL CARD                                          INPUT
300   READ (INP,3)  ICSESG,ICSECW,ICSESH,ICSECE,ICSEYM,ICSESE,ICSEAC       INPUT
     1  ,ICSEST,ICSECO,ICSEYS,RSEED1,RSEED2,LBL                            INPUT
 3    FORMAT (10I4,8X,2D12.0,2X,A6)                                        INPUT
      WRITE (OUTP,13) ICSESG,ICSECW,ICSESH,ICSECE,ICSEYM,ICSESE,ICSEAC     INPUT
     1  ,ICSEST,ICSECO,ICSEYS,RSEED1,RSEED2,LBL                            INPUT
13    FORMAT (55H0ICSESG  ICSECW  ICSESH  ICSECE  ICSEYM  ICSESE  ICSEACINPUT
     1  ,24H  ICSEST  ICSECO  ICSEYS/I6,9I8/                               INPUT
     1 8X,6HRSEED1,13X,6HRSEED2,10X,5HLABEL/2D21.12,3X,A6)                 INPUT
C         CHECK LABEL AND SEQUENCE NUMBER OF THIRD CONTROL CARD            INPUT
      IF ( LBL .EQ. 6HLEM 03 )  GO TO 400                                  INPUT
      ARG(1)= LBL                                                          INPUT
      NARG= 1                                                              INPUT
      CALL ERRMES (3HLEM,5HINPUT,20,1)                                     INPUT
C                                                                          INPUT
C         READ FOURTH CONTROL CARD                                         INPUT
400   READ (INP,4)  RSEED3,RSEED4,RSEED5,RSEED6,RSEED7,LBL                 INPUT
 4    FORMAT (5D12.0,14X,A6)                                               INPUT
      WRITE (OUTP,14) RSEED3,RSEED4,RSEED5,RSEED6,RSEED7,LBL               INPUT
14    FORMAT (/8X,6HRSEED3,14X,6HRSEED4,14X,6HRSEED5,14X,6HRSEED6,14X      INPUT
     1  ,6HRSEED7,10X,5HLABEL/5D20.12,3X,A6)                               INPUT
C         CHECK LABEL AND SEQUENCE NUMBER OF FOURTH CONTROL CARD           INPUT
      IF ( LBL .EQ. 6HLEM 04 )  GO TO 1000                                 INPUT
      ARG(1)= LBL                                                          INPUT
      NARG= 1                                                              INPUT
      CALL ERRMES (3HLEM,5HINPUT,20,1)                                     INPUT
C                                                                          INPUT
C         CHECK LEM CONTROL CARD DATA FOR ERRORS. ALSO CHECK INPUT FILE    INPUT
C         HEADERS.                                                         INPUT
1000  CALL INPCHK                                                          INPUT
C                                                                          INPUT
```

```
C        READ AND CHECK CAMS CONTROL CARDS                                  INPUT
      CALL CAMSIN                                                           INPUT
C        READ AND CHECK CAS CONTROL CARDS                                   INPUT
      CALL CASIN                                                            INPUT
C                                                                           INPUT
      CALL PAGER (5)                                                        INPUT
      WRITE (OUTP,20)  NERRS,NFATAL                                         INPUT
   20 FORMAT (//5X,I5,43H  NONFATAL ERRORS DETECTED ON CONTROL CARDS//      INPUT
     1  5X,I5,43H     FATAL ERRORS DETECTED ON CONTROL CARDS)               INPUT
  900 RETURN                                                                INPUT
      END                                                                   INPUT
```

```
FOR,IS LEM
C     PROGRAM LEM
C                                                                       LEM
C         MAIN DRIVER FOR LACIE ERROR MODEL (LEM)                      LEM
C                                                                       LEM
C         CODED BY J. R. TAYLOR, JANUARY 1976                          LEM
C                                                                       LEM
C         COMMON BLOCK DEFINITIONS                                      LEM
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         CAMS CONTROL CARD INPUT DATA                                  CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,       CAMSCM
     1     IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                   CAMSCM
      REAL MS                                                           CAMSCM
C                                                                       CAMSCM
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH     ,TOPT   ,AUNITS,DISTFF,BWIND(4)    CASCM
     3  ,WPRIOR(4)     ,APREP ,IPRD(3,14)    ,NPDATE,PRDATE(14)         CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C         DATA BLOCK FOR CAS CUMULATIVE FILE                            CASCUM
C         CAS DATA SETS 14, 15, 16, AND 17                              CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C         DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)            CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFR )                                     CASDSB
```

```
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1  , ( PIK,CASDSB(121) )                                           CASDSB
C                                                                       CASDSB
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           LEM
C        ( CHECK LISTING OF CAS FOR PROPER LENGTH OF COMMON BLOCK )     LEM
      COMMON  /CASFLG/                                                  LEM
     1   CASFLG(40)                                                     LEM
C                                                                       LEM
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CONSTANT QUANTITIES FOR LEM PROGRAM .                          CONST
      COMMON  /CONST /                                                  CONST
     1   NTRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                    IXCASF
      COMMON  /IXCASF/                                                  IXCASF
     1  IXCASF(1),LIXCAS
C                                                                       IXCASF
C        INDEX RECORD FOR CAS DISTRIBUTION FILE                         IXDISF
      COMMON  /IXDISF/                                                  IXDISF
     1   IXDISF(1),LIXDIS
C        NOTE... 506 ONLY ALLOWS UP TO 8 PREDICTION POINTS INCLUDING    IXDISF
C        BIOWINDOWS ( 506 = 1 + 1 + 8*63, (INDEX + HEADER + 8 PRED. PTS.)IXDISF
C                                                                       IXDISF
C        INDEX RECORD FOR INTERMEDIATE SUBSTRATA HISTORICAL DATA FILE   IXSUBH
      COMMON  /IXSUBH/                                                  IXSUBH
     1 LIXSSH,IXSUBH(1)                                                 MOD1
```

```
C                                                                       IXSUBH
      COMMON/FILES1/                                                    FILES1
     1ISUBH2,LSUBH2,MXCLSS                                              FILES1
C         LEM CONTROL CARD INPUT DATA                                   LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1    TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2   ,ENDR   ,ENDZ   ,ISTG   ,ICAMS ,IYES   ,IACQ   ,ICLASS,ISEXT ,ISCC   LEMCM
     3   ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4   ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5   ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                     LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1   ,RSEED6,RSEED7                                                 LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR   ,ENDZ                        LEMCM
C                                                                       LEMCM
C         PAGE EJECT CONTROL PARAMETERS FOR LEM                         PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1    NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                        PAGECM
C                                                                       PAGECM
C         STATISTICAL INFORMATION FOR LEM                               STATS
      COMMON  /STATS /                                                  STATS
     1    ITER   ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR            STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C         YIELD DATA FROM YESOUT FILE                                   YESDTA
      COMMON  /YESDTA/                                                  YESDTA
     1    YSTR   ,IZPRDD(6)     ,YSCI(6)       ,VSYCI(6)                YESDTA
     2   ,RDYES  ,NYESPP                                                YESDTA
      INTEGER  RDYES                                                    YESDTA
C                                                                       YESDTA
C                                                                       LEM
C         LOCAL VARIABLES                                               LEM
C      I       = INDEX IN MONTE CARLO DO LOOP                           LEM
C                                                                       LEM
C         LINKAGE ...    LEM IS CALLED DIRECTLY BY THE OPERATING SYSTEM  LEM
                                                                        LEM
C         SUBROUTINES USED ...                                          LEM
C            START  , INPUT  , ERRMES, INIT   , ERRMC  , SETPRF, STG    ,CAMS LEM
C            YES    , CAS    , WRAPUP                                   LEM
C                                                                       LEM
```

```
C ********************************************************************* LEM
C                                                                        LEM
C                                                                        LEM
C        INITIALIZE STORAGE, FLAGS, ETC.                                 LEM
      CALL RANACF(4,0,0,0,0,0,0)
      CALL RANACF(14,0,0,0,0,0,0)
      CALL RANACF(15,0,0,0,0,0,0)
      CALL RANACF(16,0,0,0,0,0,0)
      CALL START                                                         LEM
C                                                                        LEM
C        READ AND CHECK ALL CONTROL CARD DATA                            LEM
      CALL INPUT                                                         LEM
C                                                                        LEM
C        CHECK FOR FATAL ERRORS                                          LEM
      IF ( NFATAL .NE. 0 )  CALL ERRMES (3HLEM,3HLEM,99,2)               LEM
C                                                                        LEM
C        PERFORM INITIALIZATION TASKS  (INITIALIZE RANDOM NO. SEEDS)     LEM
      CALL INIT                                                          LEM
C                                                                        LEM
C          MONTE CARLO LOOP                                              LEM
      NSTART= RSTART + 1                                                 LEM
      DO 500 I=NSTART,NTRIAL                                             LEM
      ITER= I                                                            LEM
C        ERROR MODEL CONTROL - SET RANDOM NO. SEED FOR EACH ERROR SOURCELEM
      CALL ERRMC                                                         LEM
C                                                                        LEM
C        TEST SEGMENT TRUTH FLAG                                         LEM
      IF ( ICAMS .EQ. 2 )  GO TO 300                                     LEM
      IF ( ICAMS .NE. 0 .AND. ITER .GT. 1 )  GO TO 300                   LEM
      IF ( ISTG .EQ. 0 )  GO TO 120                                      LEM
      IF ( ISTG - 2 )  110,200,110                                       LEM
C        SEGMENT TRUTH FLAG = 1 OR 3. CALL SEGMENT TRUTH GENERATOR ONLY  LEM
C        ON THE FIRST ITERATION.                                         LEM
  110 IF ( ITER .GT. 1 )  GO TO 200                                      LEM
C        SET PRINT FLAG (PRINTF) FOR STG                                 LEM
  120 CALL SETPRF (IPRINT)                                               LEM
C        CALL SEGMENT TRUTH GENERATOR                                    LEM
      CALL STG                                                           LEM
      IF ( NFATAL .NE. 0 )  GO TO 999                                    LEM
C                                                                        LEM
C        TEST CAMS ERROR FLAG                                            LEM
```

```
 200  IF ( ICAMS .EQ. 0 )  GO TO 220                                   LEM
      IF ( ICAMS - 2 )  210,300,210                                    LEM
C        CAMS ERROR FLAG = 1 OR 3. CALL CAMS ONLY ON THE FIRST ITERATIONLEM
 210  IF ( ITER .GT. 1 )  GO TO 300                                    LEM
C        SET PRINT FLAG (PRINTF) FOR CAMS                              LEM
 220  CALL SETPRF (IPRCAM)                                             LEM
      CALL CAMS                                                        LEM
      IF ( NFATAL .NE. 0 )  GO TO 999                                  LEM
C                                                                      LEM
C        TEST YES ERROR FLAG                                           LEM
 300  IF ( IYES .EQ. 0 )  GO TO 320                                    LEM
      IF ( IYES - 2 )  310,400,310                                     LEM
C        YES ERROR FLAG = 1 OR 3. CALL YES ONLY ON THE FIRST ITERATION  LEM
 310  IF ( ITER .GT. 1 )  GO TO 400                                    LEM
C        SET PRINT FLAG (PRINTF) FOR YES                               LEM
 320  CALL SETPRF (IPRYES)                                             LEM
      CALL YES                                                         LEM
      IF ( NFATAL .NE. 0 )  GO TO 999                                  LEM
C                                                                      LEM
C        SET PRINT FLAG (PRINTF) FOR CAS                               LEM
 400  CALL SETPRF (IPRCAS)                                             LEM
      CALL CAS                                                         LEM
      IF ( NFATAL .NE. 0 )  GO TO 999                                  LEM
C        END OF MONTE CARLO LOOP                                       LEM
 500  CONTINUE                                                         LEM
C                                                                      LEM
C        END OF JOB. PRINT PROGRAM STATUS AT THE END OF THE JOB.       LEM
      CALL WRAPUP                                                      LEM
      STOP                                                             LEM
C                                                                      LEM
 999  CALL ERRMES (3HLEM;3HLEM,99,0)                                   LEM
      STOP                                                             LEM
      END                                                              LEM
```

```
      SUBROUTINE LFPA(FLDA,LMO,LYR,ALFGM,DAYS)                          LFPA
C ..                                                                    LFPA
C MODULE - P,LFPA                                                       LFPA
C OCTOBER 1, 1973                                                       LFPA
C.                                                                      LFPA
C                                                                       LFPA
C  JUNE 26,1973                                                         LFPA
C  .                                                                    LFPA
C  MODULE PPLFPA - POINT TARGET PERFORMANCE PREDICTOR                   LFPA
C                                                                       LFPA
C  GIVEN  DAY,MONTH,YEAR - SUBR. LFPA RETURNS THE RIGHT ASCENSION OF    LFPA
C  GREENWICH AT MIDNIGHT OF A GIVEN DAY                                 LFPA
C                                                                       LFPA
C     REFERENCE EPOCH IS 0 HOUR 1JAN1950                                LFPA
      DATA RADIAN /57.29578/                                            LFPA
C                                                                       LFPA
C     COMPUTE DAYS IN FULL YEARS FROM EPOCH TO LYR                      LFPA
C     DAYS TO 1JAN1963 IS 4748.                                         LFPA
C                                                                       LFPA
      DAYS=4748.                                                        LFPA
C     NOTE--LYR MUST BE GREATER THAN OR EQUAL TO (19)64                 LFPA
      LASTYR=LYR-1                                                      LFPA
      DO 20 I=63,LASTYR                                                 LFPA
      KREMD=MOD((I-60),4)                                               LFPA
      IF(KREMD.GT.0) GO TO 10                                           LFPA
      DAYS=DAYS+366                                                     LFPA
      GO TO 20                                                          LFPA
   10 DAYS=DAYS+365                                                     LFPA
   20 CONTINUE                                                          LFPA
      IF(LMO-1)30,40,30                                                 LFPA
   30 IF(LMO-2)60,50,60                                                 LFPA
   40 DAYS=DAYS+FLDA-1.                                                 LFPA
      GO TO 270                                                         LFPA
   50 DAYS=DAYS+FLDA+30.                                                LFPA
      GO TO 270                                                         LFPA
   60 KDEL=MOD((LYR-60),4)                                              LFPA
      IF(KDEL.GT.0) GO TO 70                                            LFPA
      DAYS=DAYS+59.                                                     LFPA
      GO TO 80                                                          LFPA
   70 DAYS=DAYS+58.                                                     LFPA
   80 IF(LMO-3) 270,170,90                                              LFPA
   90 IF(LMO-5) 180,190,100                                             LFPA
  100 IF(LMO-7) 200,210,110                                             LFPA
  110 IF(LMO-9) 220,230,120                                             LFPA
  120 IF(LMO-11)240,250,260                                             LFPA
  170 DAYS=DAYS+FLDA                                                    LFPA
      GO TO 270                                                         LFPA
  180 DAYS=DAYS+FLDA+31.                                                LFPA
      GO TO 270                                                         LFPA
  190 DAYS=DAYS+FLDA+61.                                                LFPA
      GO TO 270                                                         LFPA
  200 DAYS=DAYS+FLDA+92.                                                LFPA
      GO TO 270                                                         LFPA
  210 DAYS=DAYS+FLDA+122.                                               LFPA
      GO TO 270                                                         LFPA
  220 DAYS=DAYS+FLDA+153.                                               LFPA
      GO TO 270                                                         LFPA
  230 DAYS=DAYS+FLDA+184.                                               LFPA
```

```
      GO TO 270                                                         LFPA
  240 DAYS=DAYS+FLDA+214.                                               LFPA
      GO TO 270                                                         LFPA
  250 DAYS=DAYS+FLDA+245.                                               LFPA
      GO TO 270                                                         LFPA
  260 DAYS=DAYS+FLDA+275.                                               LFPA
  270 CONTINUE                                                          LFPA
      ALFGM=AMOD(SNGL((.10007554203+(.98564734600)*DAYS+                LFPA
     1(2.9015D-13)*DAYS**2)),360.)                                      LFPA
      ALFGM=ALFGM/RADIAN                                                LFPA
      RETURN                                                            LFPA
      END                                                               LFPA
```

```
@ ELT MULTI,1,760427, 39049

      SUBROUTINE MULTI(TYPE,SEASON,IWIN,M)                               MULTI
C                                                                        MULTI
C   THIS SUBROUTINE CALCULATES THE MULTI-TEMPORAL ERROR FOR TRAINING     MULTI
C      SEGMENTS.                                                         MULTI
C                                                                        MULTI
C        CAMS CONTROL CARD INPUT DATA                                    CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGFX,ISKIP,ITMAX,IREP,IWIND,        CAMSCM
     1    IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                     CAMSCM
      REAL MS                                                            CAMSCM
C                                                                        CAMSCM
      COMMON/ERROR/TITL(4),IDATE,PESTIM,TOT,ALOCAL,ERTOT(3)              ERROR
     1   ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,        ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                          ERROR
      DIMENSION IERS(40)                                                 ERROR
      EQUIVALENCE(TITL,IERS)                                             ERROR
      REAL MULT                                                          ERROR
      INTEGER TID,CROPD                                                  ERROR
      INTEGER TYPE,SEASON                                                MULTI
      DIMENSION IWIN(4),INDEX(16)                                        MULTI
      REAL M                                                             MULTI
      DATA INDEX/0,1,2,5,3,6,8,11,4,7,9,12,10,13,14,15/                  MULTI
      IFIND=1+IWIN(1)+2*IWIN(2) +4*IWIN(3) +8*IWIN(4)                    MULTI
      ISTATE=INDEX(IFIND)                                                MULTI
      IWHATM=IGROUP(TYPE,SEASON,ISTATE)                                  MULTI
      M=MS(TYPE,SEASON,IWHATM)                                           MULTI
C   ERROR REPORT CALCULATION                                             MULTI
   10 CONTINUE                                                           MULTI
      MULT(TYPE)=M                                                       MULTI
      RETURN                                                             MULTI
      END                                                                MULTI
```

```
@ FLT PAGER,1,760427, 39016
      SUBROUTINE PAGER (NLINES)                                            PAGER
C        AUTOMATIC PAGING SUBROUTINE.  PERMITS A MAXIMUM OF MXLINE         PAGER
C        LINES PER PAGE.                                                   PAGER
C                                                                          PAGER
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                             PAGECM
      COMMON  /PAGECM/                                                     PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                            PAGECM
C                                                                          PAGECM
C                                                                          PAGER
C                                                                          PAGER
      NLINE= NLINE + NLINES                                                PAGER
      IF ( NLINE .LE. MXLINE )  GO TO 900                                  PAGER
C                                                                          PAGER
      CALL EJECT (NLINES)                                                  PAGER
  900 RETURN                                                               PAGER
      END                                                                  PAGER
```

```
@ ELT PSUB,1,760427, 39092     , 1

                  FUNCTION PSUB (XX)                                              PSUB
              C      CALCULATES FUNCTION P(X) FOR CONFIDENCE LEVEL CALCULATIONS   PSUB
              C                                                                   PSUB
                  X= ABS(XX)                                                      PSUB
              C     IF ABS (X) .GT. 1.E20, THEN SET P(X) = 0 IF X IS NEGATIVE     PSUB
              C      OR P(X) = 1 IF X IS POSITIVE.                                PSUB
                   PX = 0.0                                                       PSUB
                  IF ( X .GT. 1.E20 )  GO TO 800                                  PSUB
                  PX= ((( 0.019527*X + 0.000344 )*X + 0.115194 )*X + 0.196854 )*X PSUB
                 1  + 1.0                                                         PSUB
                  PX= 0.5/ PX**4                                                  PSUB
              800 IF ( XX .GT. 0.0 )  PX= 1.0 - PX                                PSUB
                  PSUB= PX                                                        PSUB
              900 RETURN                                                          PSUB
                  END                                                             PSUB
```

```
FOR,IS RANACF
      SUBROUTINE RANACF (IFILE,IREC,BUF,N,IX,L,IOPT)                    RANACF
C        STANDARDIZED RANDOM ACCESS I/O ROUTINE                         RANACF
C        OPENS, CLOSES, READS, AND WRITES A RANDOM ACCESS FILE USING    RANACF
C        INSTALLATION DEPENDENT RANDOM ACCESS ROUTINES                  RANACF
C        ON THE CDC 6600 COMPUTING SYSTEM AT TRW THE OPENMS,CLOSEMS,    RANACF
C        READMS, AND WRITMS ENTRY POINTS ARE USED TO ACTUALLY PERFORM   RANACF
C        THE RANDOM ACCESS I/O OPERATIONS                               RANACF
C                                                                       RANACF
C        INPUT PARAMETERS ...                                           RANACF
C        IFILE = LOGICAL UNIT NUMBER OF THE RANDOM ACCESS FILE,         RANACF
C        IREC  = RECORD NUMBER TO BE READ OR WRITTEN,                   RANACF
C        BUF   = ARRAY OF LENGTH N TO CONTAIN THE RECORD,               RANACF
C     (BUF IS INPUT IF IOPT = 2 AND IS OUTPUT IF IOPT = 1)              RANACF
C        N     = LENGTH OF RECORD TO BE READ OR WRITTEN,                RANACF
C        IX    = ARRAY OF DIMENSION L CONTAINING THE INDEX RECORD,      RANACF
C        L     = LENGTH OF INDEX,                                       RANACF
C        IOPT  = ENTRY POINT OPTION ..                                  RANACF
C                = 0 TO OPEN THE FILE,                                  RANACF
C                = 1 TO READ THE FILE,                                  RANACF
C                = 2 TO WRITE THE FILE,                                 RANACF
C                = -1 TO CLOSE THE FILE.                                RANACF
C                                                                       RANACF
      DIMENSION  BUF(N),IX(L)                                           RANACF
C                                                                       RANACF
C        LINKAGE ... CALL RANACF (IFILE,IREC,BUF,N,IX,L,IOPT)           RANACF
C        RANACF IS CALLED FROM INPCHK, CAS, WRAPUP                      RANACF
C                                                                       RANACF
C        SUBROUTINES USED ... OPENMS,CLOSEMS, READMS, WRITMS            RANACF
C                                                                       RANACF
C                                                                       RANACF
C **********************************************************************RANACF
C                                                                       RANACF
      IF ( IOPT .EQ. -1 )  GO TO 400                                    RANACF
      IF ( IOPT - 1 )  100,200,300                                      RANACF
C        IOPT = 0. OPEN FILE                                            RANACF
  100 IF(IFILE .EQ. 4)DEFINE FILE 4(506,303,U,IDUM1)
      IF(IFILE .EQ. 14)DEFINE FILE 14(388,504,U,IDUM2)
      IF(IFILE .EQ. 15)DEFINE FILE 15(388,81,U,IDUM3)
      IF(IFILE .EQ. 16)DEFINE FILE 16(50,1020,U,IDUM4)
      IF(IFILE .EQ. 17)DEFINE FILE 17(320,39,U,IDUM5)
```

```
      GO TO 900                                                       RANACF
C          IOPT = 1. READ RECORD IREC                                 RANACF
 200  READ(IFILE'IREC,ERR=600)(BUF(I),I=1,N)
      GO TO 900                                                       RANACF
C          IOPT = 2. WRITE RECORD IREC                                RANACF
 300  WRITE(IFILE'IREC,ERR=600)(BUF(I),I=1,N)
      GO TO 900                                                       RANACF
C          IOPT = -1. CLOSE FILE                                      RANACF
  400 CONTINUE
      GO TO 900
  600 WRITE(6,901)IFILE,IREC
      STOP
  901 FORMAT(1H0,91H *** AN IRRECOVERABLE I/O ERROR HAS OCCURRED ON READ
     1ING A RECORD FROM A DIRECT ACCESS FILE /
     210X,6H FILE=,I5,8H RECORD=,I5,31H THE JOB IS BEING ABANDONED ***)
 900  RETURN                                                          RANACF
      END                                                             RANACF
```

@ FLT RDM1A,1,760427, 39107 , 1

```
      SUBROUTINE RDM1A(FL,U)                                            RDM1A
C     RANDOM NUMBER GENERATOR FOR ALL COMPUTERS                         RDM1A
      DOUBLE PRECISION C1,C2,R1,R2,T,FL,TWO35,ONE,ZERO,XMOD,YMOD        RDM1A
      DATA C1,C2,ONE,ZERO/3051749376.D1,84365.D0,1.D0,0.D0/,            RDM1A
     1     TWO35/34359738368.D0/                                        RDM1A
      XMOD(T) = DMOD(T,TWO35)                                           RDM1A
      YMOD(T) = DMOD(T,262144.D0)                                       RDM1A
      T = FL                                                            RDM1A
      IF(T .LE. ZERO) T = ONE                                           RDM1A
      R2 = YMOD(T)                                                      RDM1A
      R1 = T - R2                                                       RDM1A
      T = XMOD(XMOD(C1*R2+C2*R1) + C2*R2)                               RDM1A
      U = T/TWO35                                                       RDM1A
      FL = T                                                            RDM1A
      RETURN                                                            RDM1A
      END                                                               RDM1A
```

```
      SUBROUTINE REPORT(IPASS,IFIRST,IREP)                                REPORT
C                                                                         REPORT
C     THIS SUBROUTINE PRINTS THE CAMS REPORT.                             REPORT
C                                                                         REPORT
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                            PAGECM
      COMMON  /PAGECM/                                                    PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                           PAGECM
C                                                                         PAGECM
      COMMON/ERROR/TITL(4),IDATE,PLSTIM,TOT,ALOCAL,ERTOT(3)               ERROR
     1   ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,         ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                           ERROR
      DIMENSION IERS(40)                                                  ERROR
      EQUIVALENCE(TITL,IERS)                                              ERROR
      REAL MULT                                                           ERROR
      INTEGER TID,CROPD                                                   ERROR
      COMMON/SEGTRU/COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4,                SEGTRU
     1   II,IPRTOR(6),ISPW,PT(2)                                          SEGTRU
C        FILE DEFINITIONS AND RECORD LENGTHS                              FILES
      COMMON  /FILES /                                                    FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ            FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO          FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD           FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                        FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT    FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                  FILES
C                                                                         FILES
      DIMENSION IOUT(3)                                                   REPORT
    1 CONTINUE                                                            REPORT
      IF(IPASS.EQ.2) GO TO 200                                            REPORT
      IF(IPASS.EQ.0) GO TO 300                                            REPORT
C*****************************************************************        REPORT
C                                                                         REPORT
C     TRAINING SEGMENTS                                                   REPORT
      IF(IFIRST.GT.1) GO TO 10                                            REPORT
C                                                                         REPORT
C     FIRST ACQUIS. FOR SEGMENT - PRINT HEADERS                           REPORT
      CALL PAGER(7)                                                       REPORT
      WRITE(OUTP,1300)                                                    REPORT
      WRITE(OUTP,1000)                                                    REPORT
 1000 FORMAT(1X,11(10H**********))                                        REPORT
      WRITE(OUTP,1100)COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4               REPORT
 1100 FORMAT(2X,8HCOUNTRY ,A4,8H,REGION ,I2,6H,ZONE ,I3,                  REPORT
     1   8H,STRATA ,I3,11H,SUBSTRATA ,I4,18H,TRAINING SEGMENT ,I4)        REPORT
      WRITE(OUTP,1200)PT(1)                                               REPORT
 1200 FORMAT(2X,22HTRUE PROPORTION WHEAT= ,F6.2)                          REPORT
      WRITE(OUTP,1300)                                                    REPORT
 1300 FORMAT(2X)                                                          REPORT
      IF(IREP.LE.0) GO TO 20                                              REPORT
      WRITE(OUTP,1400)                                                    REPORT
 1400 FORMAT(2X,17X,9HACQ DATE ,7HESTIM. ,7HTOTAL  ,12X,                  REPORT
     1   7HERROR  ,10X,13HCLASSIF.ERROR,5X,5HCROP ,4X,6HMULTI- )          REPORT
      WRITE(OUTP,1500)                                                    REPORT
 1500 FORMAT(2X,17HCROP WINDOW      ,9HMO/DY/YR ,7HPROP.  ,               REPORT
     1   7HERROR  ,5X,2(7HWHEAT  ,7HMIXED  ,7HOTHER  ),5HCAL  ,4X,        REPORT
     1   4HTEMP)                                                          REPORT
      GO TO 30                                                            REPORT
   20 CONTINUE                                                            REPORT
      WRITE(OUTP,1600)                                                    REPORT
```

```
 1600 FORMAT(2X,17X,9HACQ DATE ,7HESTIM. ,2X,7HTOTAL   )                   REPORT
      WRITE(OUTP,1700)                                                      REPORT
 1700 FORMAT(2X,17HCROP WINDOW      ,9HMO/DY/YR ,7HPROP.  ,2X,7HERROR   )REPORT
   30 CONTINUE                                                              REPORT
   10 CONTINUE                                                              REPORT
      IF(IREP.LE.0) GO TO 40                                                REPORT
      CALL PAGER(3)                                                         REPORT
      CALL FZULU(IDATE,IOUT)                                                REPORT
      WRITE(OUTP,1010)(TITL(I),I=1,4),IOUT(2),IOUT(3),IOUT(1),PESTIM,      REPORT
     1 TOT, (ERTOT(I),I=1,3),(CLTOT(I),I=1,3),DELTA,MULT(1)                REPORT
 1010 FORMAT(2X,4A4,1X,I2,2(1H/,I2),1X,F6.2,1X,F7.2    ,5H TOT  ,          REPORT
     1    6(F6.3,1X),F5.2,2X,2HW ,F4.2)                                     REPORT
      WRITE(OUTP,1020) (ERBIAS(I),I=1,3),(CLBIAS(I),I=1,3),CROPD,          REPORT
     1  MULT(2)                                                             REPORT
 1020 FORMAT(42X,5HBIAS ,6(F6.3,1X),I3 ,4X ,2HM ,F4.2)                      REPORT
      WRITE(OUTP,1030) (ERRAND(I),I=1,3),(CLRAND(I),I=1,3),MULT(3)         REPORT
 1030 FORMAT(42X,5HRAND ,6(F6.3,1X),7X,2HO ,F4.2)                           REPORT
      RETURN                                                                REPORT
   40 CONTINUE                                                              REPORT
      CALL FZULU(IDATE,IOUT)                                                REPORT
      CALL PAGER(1)                                                         REPORT
      WRITE(OUTP,1040)(TITL(I),I=1,4),IOUT(2),IOUT(3),IOUT(1),PESTIM,TOTREPORT
 1040 FORMAT(2X,4A4,1X,I2,1H/,I2,1H/,I2,1X,F6.2,1X,F7.2)                   REPORT
      RETURN                                                                REPORT
C*****************************************************************         REPORT
C                                                                           REPORT
C   ORDINARY SEGMENTS                                                       REPORT
  200 CONTINUE                                                              REPORT
      IF(IFIRST.GT.1) GO TO 230                                             REPORT
C                                                                           REPORT
C   FIRST ACQ. FOR SEGMENT - PRINT HEADERS                                  REPORT
      CALL PAGER(7)                                                         REPORT
      WRITE(OUTP,1300)                                                      REPORT
      WRITE(OUTP,1000)                                                      REPORT
      WRITE(OUTP,2100)COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4                 REPORT
 2100 FORMAT(2X,8HCOUNTRY  ,A4,8H,REGION  ,I2,6H,ZONE  ,I3,                 REPORT
     1  8H,STRATA  ,I3,11H,SUBSTRATA  ,I4,18H,ORDINARY SEGMENT  ,I4)      REPORT
      WRITE(OUTP,1200)PT(1)                                                 REPORT
      WRITE(OUTP,1300)                                                      REPORT
  210 CONTINUE                                                              REPORT
      IF(IREP.LE.0) GO TO 220                                               REPORT
      WRITE(OUTP,2400)                                                      REPORT
 2400 FORMAT(2X,17X,9HACQ DATE ,7HESTIM. ,7HTOTAL  ,12X,                    REPORT
     1    7HERROR  ,10X,10HSIG.EXT.  ,22X,5HTRAIN)                          REPORT
      WRITE(OUTP,2500)                                                      REPORT
 2500 FORMAT(2X,17HCROP WINDOW      ,9HMO/DY/YR ,7HPROP.  ,                 REPORT
     1    7HERROR  ,5X,2(7HWHEAT  ,7HMIXED  ,7HOTHER  ),2X,6HZ1    ,       REPORT
     1    6HZ2    ,4HSEG.)                                                  REPORT
      GO TO 230                                                             REPORT
  220 CONTINUE                                                              REPORT
      WRITE(OUTP,1600)                                                      REPORT
      WRITE(OUTP,1700)                                                      REPORT
  230 CONTINUE                                                              REPORT
C                                                                           REPORT
C   PRINT INFO                                                              REPORT
      IF(IREP.LE.0) GO TO 240                                               REPORT
      CALL PAGER(3)                                                         REPORT
      CALL FZULU(IDATE,IOUT)                                                REPORT
      WRITE(OUTP,2010)(TITL(I),I=1,4),IOUT(2),IOUT(3),IOUT(1),PESTIM,      REPORT
     1  TOT, (ERTOT(I),I=1,3),(CLTOT(I),I=1,3),(Z(1,I),I=1,2),IID          REPORT
```

```
      2010 FORMAT(2X,4A4,1X,I2,2(1H/,I2),1X,F6.2,1X,F7.2, ,5H TOT  ,        REPORT
          1   6(F6.3,1X),2HW ,2(F5.2,1X),I4)                                 REPORT
           WRITE(OUTP,2020) (ERBIAS(I),I=1,3),(CLBIAS(I),I=1,3),             REPORT
          1  (Z(2,I),I=1,2),TRAINA                                           REPORT
      2020 FORMAT(42X,5HBIAS ,6(F6.3,1X),2HM ,2(F5.2,1X),F6.2)               REPORT
           WRITE(OUTP,2030)ALUCAL, (ERRAND(I),I=1,3),(CLRAND(I),I=1,3),      REPORT
          1  (Z(3,I),I=1,2),TRAIND                                           REPORT
      2030 FORMAT(35X,F6.2,1X,5HRAND ,6(F6.3,1X),2HO ,2(F5.2,1X),F6.2)       REPORT
           RETURN                                                            REPORT
       240 CONTINUE                                                          REPORT
           CALL PAGER(1)                                                     REPORT
           CALL FZULU(IDATE,IOUT)                                            REPORT
           WRITE(OUTP,1040)(TITL(I),I=1,4),IOUT(2),IOUT(3),IOUT(1),PESTIM,TOTREPORT
           RETURN                                                            REPORT
      C***************************************************************       REPORT
      C                                                                      REPORT
      C   SPECIAL CASE - NO ACQUIS. FILE                                     REPORT
       300 CONTINUE                                                          REPORT
           IF(IFIRST.GT.1) GO TO 310                                         REPORT
           CALL PAGER(7)                                                     REPORT
            WRITE(OUTP,1300)                                                 REPORT
           WRITE(OUTP,1000)                                                  REPORT
           WRITE(OUTP,3100)COUN4,IREG4,IZONE4,ISTRA4,ISUB4,ISEG4             REPORT
      3100 FORMAT(2X,8HCOUNTRY ,A4,8H,REGION ,I2,6H,ZONE ,I3,                REPORT
          1   8H,STRATA ,I3,11H,SUBSTRATA ,I4,9H,SEGMENT  ,I4)               REPORT
           WRITE(OUTP,1200)PT(1)                                             REPORT
           WRITE(OUTP,1300)                                                  REPORT
           WRITE(OUTP,1600)                                                  REPORT
           WRITE(OUTP,1700)                                                  REPORT
       310 CONTINUE                                                          REPORT
           CALL PAGER(1)                                                     REPORT
           CALL FZULU(IDATE,IOUT)                                            REPORT
           WRITE(OUTP,1040)(TITL(I),I=1,4),IOUT(2),IOUT(3),IOUT(1),PESTIM,TOTREPORT
           RETURN                                                            REPORT
           END                                                               REPORT
```

```
      SUBROUTINE RWCASF (IREC,DSLOC,IRDWR)                              RWCASF
C        READS A DATA SET FROM THE CAS CUMULATIVE FILE OR               RWCASF
C        WRITES A DATA SET ONTO THE CAS CUMULATIVE FILE.                RWCASF
C                                                                       RWCASF
C     CALLING SEQUENCE PARAMETERS ...                                   RWCASF
C        IREC  = RECORD NUMBER TO READ/WRITE                            RWCASF
C        DSLOC = ORIGIN OF DATA SET TO READ/WRITE                       RWCASF
C        IRDWR = READ/WRITE FLAG (=1 TO READ, =2 TO WRITE)              RWCASF
C                                                                       RWCASF
C        WHEN READING CASF, THE DATA IS FIRST READ INTO A BUFFER.       RWCASF
C        THEN THE DATA FOR THE PROPER PREDICTION POINT IS MOVED INTO    RWCASF
C        THE ORIGIN SPECIFIED BY DSLOC.                                 RWCASF
C                                                                       RWCASF
C        WHEN WRITING CASF, THE DATA IS FIRST MOVED INTO THE BUFFER.    RWCASF
C        AND THEN IS WRITTEN ONTO THE CAS CUMULATIVE FILE.              RWCASF
C                                                                       RWCASF
      DIMENSION  DSLOC(28)                                              RWCASF
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON /CASCUM/                                                   CASCUM
     1  CASCUM(32),   BUFFER(504)                                       CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON /CASFLG/                                                   CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYLSSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCWUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                    IXCASF
      COMMON  /IXCASF/                                                  IXCASF
     1   IXCASF(  1)  ,LIXCAS                                           IXCASF
C                                                                       IXCASF
C                                                                       RWCASF
C        SET POINTER TO PICK UP PROPER DATA SET FROM BUFFER             RWCASF
      L= (IPP - 1)*28 + 1                                               RWCASF
      IRW= IRDWR                                                        RWCASF
      IF ( IRW .EQ. 1 )  GO TO 300                                      RWCASF
C                                                                       RWCASF
C        MOVE DATA SET INTO BUFFER.                                     RWCASF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 210 I=1,28                                                     RWCASF
      BUFFR(L)= DSLOC(I)                                                RWCASF
  210 L= L + 1                                                          RWCASF
C                                                                       RWCASF
C        NOW READ OR WRITE DATA FROM OR ONTO RANDOM ACCESS FILE.        RWCASF
  300 CALL RANACF (CASF,IREC,BUFFR,LCASF,IXCASF,LIXCAS,IRW)             RWCASF
C                                                                       RWCASF
      IF ( IRW .EQ. 2 )  GO TO 900                                      RWCASF
C                                                                       RWCASF
C        DATA WAS READ FROM FILE. NOW MOVE DATA FROM BUFFER INTO        RWCASF
C        PROPER DATA SET.                                               RWCASF
      DO 310 I=1,28                                                     RWCASF
      DSLOC(I)= BUFFR(L)                                                RWCASF
  310 L= L + 1                                                          RWCASF
C                                                                       RWCASF
  900 RETURN                                                            RWCASF
      END                                                               RWCASF
```

@ ELT RWDISF,1,760427, 39095 , 1

```
      SUBROUTINE RWDISF (ILEVEL,DSET)                                   RWDISF
C        READS AND WRITES DATA FROM/ONTO THE CAS DISTRIBUTION FILE.     RWDISF
C                                                                       RWDISF
C     CALLING SEQUENCE PARAMETERS ...                                   RWDISF
C        LEVEL = 0 FOR COUNTRY                                          RWDISF
C              = 1 FOR REGION                                           RWDISF
C              = 2 FOR ZONE                                             RWDISF
C                                                                       RWDISF
C        DSET  = DATA SET 13 FOR COUNTRY                                RWDISF
C              = DATA SET 12 FOR REGION                                 RWDISF
C              = DATA SET 11 FOR ZONE                                   RWDISF
C                                                                       RWDISF
      DIMENSION  DSET(19)                                               RWDISF
C                                                                       RWDISF
C     COMMON BLOCKS                                                     RWDISF
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                      CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF    ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)  CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT   ,AUNITS,DISTFF,BWIND(4)     CASCM
     3  ,WPPIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C        DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)             CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFR )                                     CASDSB
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1  , ( PIK,CASDSB(121) )                                           CASDSB
C                                                                       CASDSB
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW    ,IBW    ,WINDOW,IPD    ,IPP    ,PPDATE,NREGS CASFLG
     2  ,NZTOT  ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES  ,NRSSH  ,NRCAMS     CASFLG
     3  ,ENDC   ,ENDRLG,ENDZON,IRSTR  ,IRZONE,IRREG                     CASFLG
     4  ,LDS1   ,LDS4  ,LDS7   ,LDS8   ,LDS9  ,LDS10  ,LDS11 ,LDS12 ,LDS13 CASFLG
     5  ,LDS14  ,LDS15 ,LDS16  ,LDS17  ,LRCOUN,LRREG ,LRZONE,LRSTR     CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YLSERR,LYLSER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1  ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF               FILES
```

```
C                                                                       FILES
C       DATA BLOCK FOR CAS DISTRIBUTION FILE                           IXDISF
      COMMON  /IXDISF/                                                  IXDISF
     1   IXDISF(  1), LIXDIS                                            IXDISF
C        NOTE... 506 ONLY ALLOWS UP TO 8 PREDICTION POINTS INCLUDING    IXDISF
C        BIOWINDOWS ( 506 = 1 + 1 + 8*63, INDEX + HEADER + 8 PRED. PTS.)IXDISF
C                                                                       IXDISF
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYLSR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C                                                                       RWDISF
C     NOTE ... THERE ARE 9 COUNTRY VALUES PER ITERATION PER PREDICTION  RWDISF
C              POINT REQUIRING 3 COUNTRY RECORDS PER PREDICTION POINT.  RWDISF
C              (100 ITERATIONS/RECORD)                                  RWDISF
C              THERE ARE 3 REGION OR ZONE VALUES PER ITERATION PER      RWDISF
C              PREDICTION POINT REQUIRING 1 REGION OR ZONE RECORD PER   RWDISF
C              PREDICTION POINT.                                        RWDISF
C              (100 ITERATIONS/RECORD)                                  RWDISF
C                                                                       RWDISF
C          3 COUNTRY RECORDS + 10 REGION RECORDS + 50 ZONE RECORDS =    RWDISF
C          63 RECORDS PER PREDICTION POINT.                             RWDISF
C                                                                       RWDISF
C                                                                       RWDISF
      LEVEL= ILEVEL                                                     RWDISF
C                                                                       RWDISF
C        FOR THE IPP-TH PREDICTION POINT THE 3 COUNTRY RECORDS ARE ...  RWDISF
C        2+N, 3+N, 4+N WHERE N = (IPP-1)*63                             RWDISF
      IREC= 2 + (IPP-1)*63                                              RWDISF
C                                                                       RWDISF
C        THE 10 REGION RECORDS ARE 5+N, 6+N, ..., 14+N                  RWDISF
      IF ( LEVEL .EQ. 1 )  IREC= IREC + IRREG                           RWDISF
C                                                                       RWDISF
C        THE 50 ZONE RECORDS ARE 15+N, 16+N, ... , 64+N                 RWDISF
      IF ( LEVEL .EQ. 2 )  IREC= IREC + IRZONE                          RWDISF
C                                                                       RWDISF
      IF ( NT .GT. 1 )  GO TO 120                                       RWDISF
C        FIRST ITERATION.  CLEAR BUFFER AND STORE REFERENCE VALUES      RWDISF
      DO 110 I=1,LCASD                                                  RWDISF
      CASDSB(I)= 0.0                                                    RWDISF
  110 CONTINUE                                                          RWDISF
C                                                                       RWDISF
C        AREA REFERENCE VALUE = TRUE WA                                 RWDISF
      CASDSB(1)= DSET(2)                                                RWDISF
C        PRODUCTION REFERENCE VALUE = TRUE PRODUCTION                   RWDISF
      CASDSB(2)= DSET(6)                                                RWDISF
C        YIELD REFERENCE VALUE = 100.0                                  RWDISF
      CASDSB(3)= 100.0                                                  RWDISF
      GO TO 130                                                         RWDISF
C                                                                       RWDISF
  120 IF ( NT .LE. 100 )  GO TO 125                                     RWDISF
      CALL ERRMES (3HCAS,6HRWDISF,17,1)                                 RWDISF
      DISTFF= 0                                                         RWDISF
      GO TO 900                                                         RWDISF
C                                                                       RWDISF
C        READ DATA RECORD INTO BUFFER                                   RWDISF
  125 CALL RAMACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,1)            RWDISF
C                                                                       RWDISF
C        STORE AREA ERROR, PRODUCTION ERROR, AND YIELD ERROR FOR        RWDISF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C          ITERATION NT                                                  RWDISF
        130   CASDSB(NT+3)= DSET(4)                                            RWDISF
              CASDSB(NT+103)= DSET(8)                                          RWDISF
              CASDSB(NT+203)= DSET(12)                                         RWDISF
      C                                                                        RWDISF
      C          WRITE RECORD BACK ONTO CAS DISTRIBUTION FILE.                 RWDISF
              CALL RANACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,2)           RWDISF
      C                                                                        RWDISF
              IF ( LEVEL .NE. 0 )  GO TO 900                                   RWDISF
      C                                                                        RWDISF
      C          PROCESS SECOND COUNTRY RECORD                                 RWDISF
              IREC= IREC + 1                                                   RWDISF
              IF ( NT .GT. 1 )  GO TO 220                                      RWDISF
      C          STORE REFERENCE VALUES FOR SECOND COUNTRY RECORD.             RWDISF
              CASDSB(1)= 100.0                                                 RWDISF
              CASDSB(2)= 100.0                                                 RWDISF
              GO TO 230                                                        RWDISF
      C                                                                        RWDISF
      C          READ RECORD INTO BUFFER                                       RWDISF
        220   CALL RANACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,1)           RWDISF
      C                                                                        RWDISF
      C          STORE CLFWA, CLEPRD, AND CLATEC FOR ITERATION NT.             RWDISF
        230   CASDSB(NT+3)= DSET(20)                                           RWDISF
              CASDSB(NT+103)= DSET(21)                                         RWDISF
              CASDSB(NT+203)= DSET(22)                                         RWDISF
      C                                                                        RWDISF
      C          WRITE RECORD BACK ONTO CAS DISTRIBUTION FILE.                 RWDISF
              CALL RANACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,2)           RWDISF
      C                                                                        RWDISF
      C          PROCESS THIRD COUNTRY RECORD                                  RWDISF
              IREC= IREC + 1                                                   RWDISF
              IF ( NT .EQ. 1 )  GO TO 330                                      RWDISF
      C          REFERENCE VALUES ALREADY STORED IN CASDSB.                    RWDISF
      C                                                                        RWDISF
      C          READ DATA RECORD INTO BUFFER                                  RWDISF
              CALL RANACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,1)           RWDISF
      C                                                                        RWDISF
      C          STORE CLPTEC, CLATWC, AND CLPTWC FOR ITERATION NT.            RWDISF
        330   CASDSB(NT+3)= DSET(23)                                           RWDISF
              CASDSB(NT+103)= DSET(24)                                         RWDISF
              CASDSB(NT+203)= DSET(25)                                         RWDISF
      C                                                                        RWDISF
      C          WRITE RECORD BACK ONTO CAS DISTRIBUTION FILE.                 RWDISF
              CALL RANACF (CASDIS,IREC,CASDSB,LCASD,IXDISF,LIXDIS,2)           RWDISF
      C                                                                        RWDISF
        900   RETURN                                                           RWDISF
              END                                                              RWDISF
```

```
 FOR,IS SEGTAB
      SUBROUTINE SEGTAB                                                 SEGTAB
C     THIS ROUTINE FORMS THE SEGMENT TABLES TO BE USED TO DETERMINE     SEGTAB
C     CLASS                                                             SEGTAB
C        TABLES NECCESSARY TO DETERMINE CLASS SETS WITHIN A ZONE        CLSTAB
      COMMON /CLSTAB/                                                   CLSTAB
     1       ISTRAT(300),ISBSTR(300),NSCNT(300),IGROUP(300),IDAT1(300), MOD1
     2       IDAT2(300),XORD(300),IXPT(300),IRANK(300),IBPT(10),IEPT(10),MOD1
     3       MAXCLS,ICLCNT,ISUB1,NAGQ                                   CLSTAB
      DIMENSION DAT1(300),DAT2(300),RANK(300)                           MOD1
      EQUIVALENCE (IDAT1(1),DAT1(1)),(IDAT2(1),DAT2(1)),(IRANK(1),      CLSTAB
     1RANK(1))                                                          CLSTAB
      DIMENSION IGAP(300),GAP(300)                                      MOD1
      EQUIVALENCE (GAP(1),IGAP(1))                                      SEGTAB
      DATA IMAX/300/,XCON/10.E20/                                       MOD1
      IPT = 0                                                           SEGTAB
      DO 5 I=1,IMAX                                                     SEGTAB
      IGAP(I) = 0                                                       SEGTAB
      XORD(I) = 0.0                                                     SEGTAB
      IXPT(I) = 0                                                       SEGTAB
      IRANK(I) = 0                                                      SEGTAB
    5 CONTINUE                                                          SEGTAB
      ICT = 1                                                           SEGTAB
      DO 30 I=1,ISUB1                                                   SEGTAB
      IF(NSCNT(I) .EQ. 0)GO TO 30                                       SEGTAB
      ITEM = NSCNT(I)                                                   SEGTAB
      DO 15 J=1,ITEM                                                    SEGTAB
      IPT = IPT + 1                                                     SEGTAB
      IGAP(IPT) = IDAT1(I)                                              SEGTAB
      IRANK(IPT) = IDAT2(I)                                             SEGTAB
   15 CONTINUE                                                          SEGTAB
      IXPT(I) = ICT                                                     SEGTAB
      ICT = ICT + NSCNT(I)                                              SEGTAB
   30 CONTINUE                                                          SEGTAB
      SUM = 0.0                                                         SEGTAB
      DO 35 I=1,IPT                                                     SEGTAB
      DAT2(I) = GAP(I)*SQRT(RANK(I)*(1. - RANK(I)))                     MOD1
      SUM = SUM + DAT2(I)                                               SEGTAB
   35 CONTINUE                                                          SEGTAB
      DO 40 I=1,IPT                                                     SEGTAB
      RANK(I) = DAT2(I)/SUM                                             SEGTAB
```

```
      XORD(I) = RANK(I)                                                 SEGTAB
   40 CONTINUE                                                          SEGTAB
      DO 45 I=1,ISUB1                                                   SEGTAB
      IDAT1(I) = IXPT(I)                                                SEGTAB
   45 CONTINUE                                                          SEGTAB
      DO 50 I=1,IPT                                                     SEGTAB
      IXPT(I) = I                                                       SEGTAB
   50 CONTINUE                                                          SEGTAB
      CALL SORTAG(RANK,1,IPT,IXPT)                                      SEGTAB
      DO 55 I=1,IPT                                                     SEGTAB
      GAP(I) = 0.0                                                      SEGTAB
   55 CONTINUE                                                          SEGTAB
      IPT1 = IPT - 1                                                    SEGTAB
      DO 60 I=1,IPT1                                                    SEGTAB
      GAP(I) = RANK(I+1) - RANK(I)                                      SEGTAB
   60 CONTINUE                                                          SEGTAB
      IRK = 0                                                           SEGTAB
      XMINS = XCON                                                      SEGTAB
      DO 90 I=1,IPT1                                                    SEGTAB
   65 XMIN = XCON                                                       SEGTAB
      DO 80 J=1,IPT1                                                    SEGTAB
      IF(GAP(J) .LT. 0.)GO TO 80                                        SEGTAB
      IF(GAP(J) .GT. XMIN)GO TO 80                                      SEGTAB
      XMIN = GAP(J)                                                     SEGTAB
      IMPT = J                                                          SEGTAB
   80 CONTINUE                                                          SEGTAB
      IF(XMIN .NE. XMINS)IRK = IRK + 1                                  SEGTAB
      XMIN = GAP(IMPT)                                                  SEGTAB
      GAP(IMPT) = -10.                                                  SEGTAB
      IRANK(IMPT) = IRK                                                 SEGTAB
      XMINS = XMIN                                                      SEGTAB
   90 CONTINUE                                                          SEGTAB
  100 RETURN                                                            SEGTAB
      END                                                               SEGTAB
```

```
      SUBROUTINE SETPRF (IPR)                                           SETPRF
C        SETS THE PRINT FLAG  PRINTF TO PRINT REPORTS OR SUPPRESS       SETPRF
C        PRINTING FOR A GIVEN MODULE DEPENDING UPON THE ITERATION NUMBERSETPRF
C        AND THE INPUT PRINT FLAG FOR THAT MODULE                       SETPRF
C                                                                       SETPRF
C        INPUT PARAMETER ...                                            SETPRF
C          IPR = 0 TO PRINT REPORTS ON FIRST AND LAST ITERATIONS,       SETPRF
C          IPR = 1 TO PRINT REPORTS ON EVERY ITERATION,                 SETPRF
C          IPR = 2 TO PRINT REPORTS ONLY ON THE LAST ITERATION,         SETPRF
C          IPR = 3 TO SUPPRESS PRINTING.                                SETPRF
C                                                                       SETPRF
C        COMMON BLOCK DEFINITIONS                                       SETPRF
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(1)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ   ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NI,ITER )                                          STATS
C                                                                       STATS
C                                                                       SETPRF
C        LINKAGE ...    CALL SETPRF (IPR)                               SETPRF
C        SETPRF IS CALLED FROM THE LEM DRIVER                           SETPRF
C                                                                       SETPRF
C ********************************************************************* SETPRF
C                                                                       SETPRF
C                                                                       SETPRF
C        INITIALLY TURN PRINT FLAG OFF.                                 SETPRF
      PRINTF= 0                                                         SETPRF
C        TEST INPUT PRINT FLAG                                          SETPRF
      IF ( IPR .EQ. 3 )  GO TO 90                                       SETPRF
      IF ( IPR - 1 )  10,30,20                                          SETPRF
C        IPR = 0, PRINT ON FIRST AND LAST ITERATIONS OF THIS RUN        SETPRF
   10 IF ( ITER .EQ. NSTART )  GO TO 30                                 SETPRF
C        IPR = 2  OR IPR = 0,AND NOT FIRST ITERATION                    SETPRF
   20 IF ( ITER .NE. NTRIAL )  GO TO 90                                 SETPRF
C        IPR = 1  OR IPR= 0 AND FIRST OR LAST ITERATION OR IPR = 2 AND  SETPRF
C        LAST ITERATION                                                 SETPRF
   30 PRINTF= 1                                                         SETPRF
   90 RETURN                                                            SETPRF
      END                                                               SETPRF
```

@ ELT SGEXT,1,760427, 39051   , 1

```
      SUBROUTINE  SGEXT(SEED3,TYPE,WINDOW,IUSF,XI)                       STGFXT
C                                                                        STGFXT
C  THIS SUBROUTINE CALCULATES THE SIGNATURE EXTENSION ERROR FOR          STGFXT
C     ORDINARY SEGMENTS.                                                 STGFXT
C                                                                        STGEXT
      COMMON/CAMERR/ COUN2,IRFG2,IZONF2,ISTRA2,ISUB2,ISFG2,              CAMERR
     1  PW(3,4),BFRR(3,4),SIGERR(3,4)                                    CAMERR
      COMMON/TRAINS/  COUN7,IRFG7,IZONE7,ISTRA7,ISUB7,ISEG7,             TRAINS
     1   TIWIN(4,25),ITTOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),          TRAINS
     1   TPTRUE,TIZULU(4),TPEST(4),TPERR(4),TERTOT(3),TM(3),TV(3),TB(3)  TRAINS
      INTEGER TIZULU                                                     TRAINS
      DIMENSION ITRAIN(129)                                              TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                          TRAINS
      COMMON/SIGEX/COUN5,IRFG5,IZONF5,ZB(3,2),ZSIG(3,2,6)                SIGFX
C        CONTROL PARAMETERS FOR LEN PROGRAM                              CNTRL
      COMMON  /CNTRL /                                                   CNTRL
     1   PRINTF,NSTART,SEED(7)                                           CNTRL
      INTEGER  PRINTF                                                    CNTRL
      DOUBLE PRECISION  SEED                                             CNTRL
C                                                                        CNTRL
C        CAMS CONTROL CARD INPUT DATA                                    CAMSCM
      COMMON/CAMSCM/ IMODEL,IMULTI,ISIGEX,ISKIP,ITMAX,IREP,IWIND,        CAMSCM
     1    IGROUP(3,2,15),MS(3,2,3),G(3,2,2),H(3,2,2)                     CAMSCM
      REAL MS                                                            CAMSCM
C                                                                        CAMSCM
      COMMON/ERROR/TITL(4),TDATE,PESTIM,TOT,ALOCAL,ERTOT(3)              ERROR
     1   ,ERBIAS(3),ERRAND(3),CLTOT(3),CLBIAS(3),CLRAND(3),DELTA,        ERROR
     1  CROPD, Z(3,2),MULT(3),TID,TRAINA,TRAIND                          ERROR
      DIMENSION IERS(40)                                                 ERROR
      EQUIVALENCE(TITL,IERS)                                             ERROR
      REAL MULT                                                          ERROR
      INTEGER TID,CROPD                                                  ERROR
C        ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
      DOUBLE PRECISION SEED3                                             SIGFXT
      INTEGER TYPE, WINDOW                                               SIGEXT
      DIMENSION V(2)                                                     SIGFXT
      SIGMA=PW(TYPE,WINDOW)*ABS(TERTOT(TYPE)*ISIGEX*ZSIG(TYPE,1,IUSE)    SIGFXT
     1      +(1-ISIGEX)*ZSIG(TYPE,2,IUSE))                               SIGFXT
      XBAR=PW(TYPE,WINDOW)*  (1. +TERTOT(TYPE) *ZB(TYPE,1) +ZB(TYPE,2))  SIGFXT
      CALL BETAD(SEED3,XBAR,SIGMA,XI,0,IER)                              SIGEXT
      NARG=1                                                             SIGFXT
      IARG(1)=IER                                                        SIGEXT
      IF(IER.GT.0) CALL ERRMES(4HCAMS,5HSGEXT,4,0)                       SIGEXT
      IF(IER.GE.3) XI=XBAR                                               SIGFXT
C                                                                        SIGFXT
C  COMPUTE ERROR REPORT QUANTITIES                                       SIGFXT
   10 CONTINUE                                                           SIGFXT
      IF(PRINTF.LE.0.OR.IREP.LE.0) RETURN                                SIGFXT
      V(1)=0.                                                            SIGFXT
      V(2)=0.                                                            SIGFXT
      IF(SIGMA.LE.0.) GO TO 20                                           SIGFXT
      IF(ISIGEX.GT.1)V(1)=(XI-XBAR)/(PW(TYPE,WINDOW)*TERTOT(TYPE))       SIGFXT
      IF(ISIGEX.LT.0)V(2)=(XI-XBAR)/PW(TYPE,WINDOW)                      SIGFXT
```

ORIGINAL PAGE IS
OF POOR QUALITY

28234-6029-RU-00
Page 503

```
20 CONTINUE                                                                SIGFXT
   Z(TYPE,1)=ZB(TYPE,1)+V(1)                                               SIGFXT
   Z(TYPE,2)=ZB(TYPE,2)+V(2)                                               SIGFXT
   ERTOT(TYPE)=TERTOT(TYPE)* Z(TYPE,1)  + Z(TYPE,2)                        SIGFXT
   ERBIAS(TYPE)=TM(TYPE)*TB(TYPE)*ZB(TYPE,1)+ZB(TYPE,2)                    SIGFXT
   ERRAND(TYPE)=TM(TYPE)*(TV(TYPE)*ZB(TYPE,1) +TB(TYPE)*V(1)               SIGFXT
  1  +TV(TYPE)*V(1)) + V(2)                                                SIGFXT
   CLTOT(TYPE)=(TB(TYPE)+TV(TYPE))*Z(TYPE,1)  +Z(TYPE,2)                   SIGFXT
   CLBIAS(TYPE)=TB(TYPE)*ZB(TYPE,1) +ZB(TYPE,2)                            SIGFXT
   CLRAND(TYPE)=TV(TYPE)*ZB(TYPE,1) +TB(TYPE)*V(1)                         SIGFXT
  1      +TV(TYPE)*V(1)  +V(2)                                             SIGFXT
   RETURN                                                                  SIGFXT
   END                                                                     SIGFXT
```

```
      SUBROUTINE START                                                  START
C        JOB INITIALIZATION ROUTINE. INITIALIZES STORAGE, FLAGS,        START
C        COUNTERS, ETC.                                                 START
C                                                                       START
C        COMMON BLOCK DEFINITIONS                                       START
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                           ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C        CONTROL PARAMETERS FOR LEM PROGRAM                             CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C        CONSTANT QUANTITIES FOR LEM PROGRAM                            CONST
      COMMON  /CONST /                                                  CONST
     1   NIRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSFG                       CONST
C                                                                       CONST
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASF ,CONTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTU  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSFSG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSFST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTP,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NI,ITER )                                          STATS
C                                                                       STATS
C                                                                       START
C        LINKAGE ...   CALLED BY LEM DRIVER                             START
C                                                                       START
C                                                                       START
      NSTTL= 0                                                          START
C                                                                       START
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 110 I=1,7                                                      START
      NREC(I)= 0                                                        START
  110 CONTINUF                                                          START
C                                                                       START
      RETURN                                                            START
      END                                                               START
```

@ ELT STG,1,760430, 50105 , 1

```
      SUBROUTINE STG                                                    STG
C         SEGMENT TRUTH GENERATOR FOR THE LEM PROGRAM                   STG
C                                                                       STG
C        READS DATA FROM THE SEGMENT ID FILE AND THE SUBSTRATA HISTORICALSTG
C         FILE. CALCULATES THE TRUE PR AND TRUE PM FOR EACH SEGMENT,    STG
C         AND GENERATES THE SEGMENT TRUTH FILE                          STG
C                                                                       STG
C         BESIDES THE TWO INPUT FILES (SEGID AND SUBHST) THE FOLLOWING  STG
C         QUANTITIES ARE INPUTS TO STG ...                              STG
C         ICASE  = CASE NUMBER                                          STG
C         CUNTRY = COUNTRY                                              STG
C         NIRIAL = FINAL MONTE CARLO ITERATION FOR THIS RUN             STG
C         RSTART = INITIAL MONTE CARLO ITERATION FOR THIS RUN - 1       STG
C         STARTR = STARTING REGION                                      STG
C         STARTZ = STARTING ZONE                                        STG
C         ENDR   = ENDING REGION                                        STG
C         ENDZ   = ENDING ZONE                                          STG
C         ISTG   = SEGMENT TRUTH OPTION                                 STG
C              =0 TO VARY SEGMENT TRUTH ERROR ON EVERY ITERATION,       STG
C              =1 TO HOLD ERROR CONSTANT BY PERFORMING CALCULATIONS ONLY STG
C                 ON THE FIRST ITERATION,                               STG
C              =3 TO ELIMINATE SEGMENT TRUTH ERROR  (ERROR IS ZERO)     STG
C                 SEGMENT TRUTH FILE WILL BE WRITTEN ONLY ON THE FIRST  STG
C                 ITERATION IF ISTG = 1 OR 3                            STG
C              STG SHOULD NEVER BE CALLED IF ISTG = 2.                  STG
C         ITER   = MONTE CARLO ITERATION NUMBER                         STG
C         PRINTF = PRINT FLAG (=1 TO PRINT REPORT, =0 OTHERWISE)        STG
C         SEED(1)= RANDOM NUMBER SEED FOR SEGMENT TRUTH                 STG
C                                                                       STG
C         THE FOLLOWING OUTPUT QUANTITIES ARE STORED IN COMMON BY STG ...STG
C         NREC(1)= NO. OF DATA RECORDS PROCESSED BY STG FROM SEGID      STG
C         NREC(3)= NO. OF DATA RECORDS PROCESSED BY STG FROM SUBHST     STG
C         NSEGTR = NO. OF RECORDS WRITTEN ON THE SEGMENT TRUTH FILE.    STG
C                                                                       STG
C         COMMON BLOCK DEFINITIONS                                      STG
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         CONTROL PARAMETERS FOR LEM PROGRAM                            CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C         CONSTANT QUANTITIES FOR LEM PROGRAM                           CONST
      COMMON  /CONST /                                                  CONST
     1   NIRMX ,MAXR  ,MAXZ  ,IMXSEG,ENDFIL,ITSEG                       CONST
C                                                                       CONST
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,TMP   ,OUTP  ,TACQ  ,LTACQ ,CASOSF,LCASOS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
```

```
     1 ,STGFX1,YESERR,SFGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                  FILES
C                                                                        FILES
C        LEM CONTROL CARD INPUT DATA                                     LEMCM
      COMMON  /LEMCM /                                                   LEMCM
     1  TITLE(10)     ,ICASE  ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR   ,ENDZ   ,ISTG   ,ICAMS ,IYFS   ,IACQ   ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECM,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECD,ICSEYS,ICSECU,ICSECD                       LEMCM
      DIMENSION  RSEED(7)                                                LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5        LEMCM
     1  ,RSEED6,RSEED7                                                   LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                      LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR   ,ENDZ                         LEMCM
C                                                                        LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                           PAGECM
      COMMON  /PAGECM/                                                   PAGECM
     1  NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                           PAGECM
C                                                                        PAGECM
C        STATISTICAL INFORMATION FOR LEM                                 STATS
      COMMON  /STATS /                                                   STATS
     1  ITER  ,NSEGTR,NCAMSR,NYESR ,NRFC(7),NCASCR,NCASDR               STATS
      EQUIVALENCE  ( NI,ITER )                                           STATS
C                                                                        STATS
C        DATA FOR SEGMENT TRUTH GENERATOR                                STGDTA
      COMMON  /STGDTA/                                                   STGDTA
     1  COUN   ,IREG   ,IZONE  ,ISTRAT,ISUBS ,ISEG   ,ITRAIN,ITSPRL(6)   STGDTA
     2  ,SLAT   ,SLONG ,GRIDNO,ISW    ,COUN2 ,IREG2 ,IZONE2,ISTRA2       STGDTA
     3  ,ISUBS2,NSEG   ,IDSEG  ,GRPNO ,HISTPW,AREA   ,PWK    ,NAGR       STGDTA
     4  ,NA     ,DELTPW,DELTPM,CV1    ,CV2    ,CV3    ,CV4    ,PWKI       STGDTA
     5  ,PMKI   ,AVEPW ,SUMPW ,SNBR   ,PMMEAN,PW     ,PM     ,ERRPW      STGDTA
     6  ,SIGMA ,ERRPWT                                                   STGDTA
      DIMENSION  IDSEG(150), PW(150), PM(150), ERRPW(150)                STGDTA
      INTEGER  GRIDNO,GRPNO                                              STGDTA
C                                                                        STGDTA
C                                                                        STG
C        LOCAL VARIABLES                                                 STG
C        NAME   = ALPHANUMERIC FILENAME FOR THE SEGMENT TRUTH FILE       STG
C        RDSID  = READ FLAG FOR SEGID ( = 1 TO SKIP READING SEGID FOR    STG
C                 ONE PASS, 0 OTHERWISE.)                                STG
C        ENDD   = END-OF-DATA FLAG ( = 1 WHEN THE END OF DATA IS         STG
C                 DETECTED ON SEGID AND/OR SUBHST)                       STG
C        NSEGS  = NUMBER OF SEGMENTS FOUND ON THE CURRENT SUBSTRATA      STG
C                 (SHOULD = NSEG)                                        STG
C        IERROR= ERROR FLAG RETURNED FROM SUBR. BETAD                    STG
C        ZERO   = ZERO WORD USED TO FILL OUT SHORT RECORDS ON SEGTRU FILESTG
C        NFILL = NUMBER OF WORDS REQUIRED TO FILL OUT SHORT RECORDS      STG
C                                                                        STG
C        LINKAGE ...    CALL STG                                         STG
C        STG IS CALLED FROM THE LEM DRIVER                               STG
C                                                                        STG
C        SUBROUTINES USED ...                                            STG
C           BETAD  = BETA DISTRIBUTION ROUTINE                           STG
C           ERRMES = ERROR MESSAGE ROUTINE                               STG
C                                                                        STG
C        FILENAME FOR SEGMENT TRUTH FILE                                 STG
      DIMENSION  NAME(2)                                                 STG
      DATA  NAME / 4HSEGT,4HRUTH /,ZERO/0/
C                                                                        STG
C ********************************************************************** STG
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C                                                                   STG
      C                                                                   STG
      C     SKIP HEADER RECORDS OF SEGMENT ID FILE AND SUBSTRATA HIST. FILFSTG
            REWIND SEGID                                                  STG
            READ (SEGID)                                                  STG
            REWIND SUBHST                                                 STG
            READ (SUBHST)                                                 STG
            REWIND SEGTRU                                                 STG
      C     WRITE HEADER RECORD OF SEGMENT TRUTH FILE                     STG
            NFILL= ISEGTR - 4                                             STG
            WRITE (SEGTRU)  NAME,ICASE,IISEG,( ZERO,I=1,NFILL )           STG
      C                                                                   STG
      C     INITIALIZE FLAGS, COUNTERS, ETC.                              STG
            NREC(1)= 0                                                    STG
            NREC(3)= 0                                                    STG
            NSEGIP = 0                                                    STG
            RDSID  = 0.0                                                  STG
            ENDD   = 0.0                                                  STG
            SUMPW  = 0.0                                                  STG
            NSEGS  = 0                                                    STG
            NLINE= MXLINE                                                 STG
      C     READ DATA RECORD FROM SUBSTRATA HISTORICAL FILE               STG
      C                                                                   STG
        200 READ (SUBHST)   COUN2,IREG2,IZONE2,ISTRA2,ISUBS2,NSEG         STG
           1  , ( IDSEG(I),I=1,IMXSEG ), GRPNO,HISTPW,AREA,PWK,NAGR,NA,DELIPW STG
           2  ,DELIPM,CV1,CV2,CV3,CV4                                     STG
        220 IF (  COUN2 .EQ. ENDFIL )  GO TO 600                          STG
      C     ARE REGION AND ZONE FROM SUBHST WITHIN THE RANGE DETERMINED BY STG
      C     STARTR, STARTZ, ENDR, AND ENDZ                                STG
            IF ( IREG2 - STARTR )  200,230,240                            STG
      C     IREG2 = STARTR. NOW COMPARE IZONE2 TO STARTZ                  STG
        230 IF ( IZONE2 .LT. STARTZ )  GO TO 200                          STG
      C     IREG2 .GE. STARTR  AND  IZONE2 .GE. STARTZ                    STG
        240 IF ( ENDR .EQ. 0 )  GO TO 260                                 STG
            IF ( IREG2 - ENDR )  260,250,600                              STG
      C     IREG2 = ENDR.  NOW COMPARE IZONE2 TO ENDZ                     STG
        250 IF ( IZONE2 .GT. ENDZ )  GO TO 600                            STG
      C     IREG2 .LE. ENDR  AND  IZONE2 .LE. ENDZ                        STG
        260 IF ( NSEG .EQ. 0 )  GO TO 200                                 STG
      C     ADVANCE SUBHST COUNTER                                        STG
            NREC(3)= NREC(3) + 1                                          STG
      C                                                                   STG
      C     ARE WE READY TO READ A SEGMENT FROM THE SEGMENT ID FILE       STG
      C     ( WE MAY HAVE ALREADY READ THE FIRST SEGMENT FOR THE CURRENT  STG
      C       SUBSTRATA BEFORE READING THE SUBSTRATA FILE )               STG
            IF ( RDSID .EQ. 0.0 )  GO TO 300                              STG
      C     SET FLAG TO READ SEGMENT ID FILE NEXT TIME THROUGH THIS LOGIC STG
            RDSID= 0.0                                                    STG
            GO TO 400                                                     STG
      C                                                                   STG
      C     READ A DATA RECORD FROM THE SEGMENT ID FILE                   STG
        300 READ (SEGID)  COUN,IREG,IZONE,ISTRAT,ISUBS,ISEG,ITRAIN,ITSPRL STG
           1  ,SLAT,SLONG,GRIDNO,ISW                                      STG
        320 IF ( COUN .EQ. ENDFIL )  GO TO 600                            STG
      C     ARE REGION AND ZONE FROM SEGID WITHIN THE RANGE DETERMINED BY STG
      C     STARTR, STARTZ, ENDR, AND ENDZ.                               STG
            IF ( IREG - STARTR )  300,330,340                             STG
        330 IF ( IZONE .LT. STARTZ )  GO TO 300                           STG
      C     IREG .GE. STARTR  AND  IZONE .GE. STARTZ                      STG
        340 IF ( ENDR .EQ. 0 )  GO TO 400                                 STG
```

```
      IF ( IREG - ENDR )  400,350,600                                   STG
C         IREG = ENDR.  NOW COMPARE IZONE TO ENDZ                        STG
  350 IF ( IZONE .GT. ENDZ )  GO TO 600                                 STG
C         IREG .LE. ENDR  AND  IZONE .LE. ENDZ                           STG
C                                                                       STG
C         COMPARE SUBSTRATA FROM SEGID TO SUBSTRATA FROM SUBHST.         STG
  400 IF ( IREG   - IREG2  )  440,410,470                               STG
  410 IF ( IZONE  - IZONE2 )  440,420,470                               STG
  420 IF ( ISTRAT - ISTRA2 )  440,430,470                               STG
  430 IF ( ISUBS  - ISUBS2 )  440,480,470                               STG
C                                                                       STG
C         SUBSTRATA FROM SEGID  .LT. SUBSTRATA FROM SUBHST               STG
C         SOMETHING IS WRONG. THE FILES ARE INCONSISTENT WITH EACH OTHER STG
C         OR ONE FILE IS OUT OF ORDER. WRITE ERROR MESSAGE, DROP THIS    STG
C         SEGMENT AND CONTINUE.                                          STG
  440 IF ( ITER .EQ. RSTART + 1 )  CALL ERRMES (3HSTG,3HSTG,1,0)        STG
      GO TO 300                                                         STG
C                                                                       STG
C         SUBSTRATA FROM SEGID  .GT.  SUBSTRATA FROM SUBHST.             STG
C         THE END OF THE CURRENT SUBSTRATA FROM SUBHST HAS BEEN REACHED. STG
C         SET FLAG TO SKIP READING SEGID ON THE NEXT PASS SINCE THE      STG
C         FIRST SEGMENT OF THE NEXT SUBSTRATA HAS ALREADY BEEN READ FROM STG
C         SEGID.                                                         STG
  470 RDSTD= 1.0                                                        STG
      GO TO 610                                                         STG
C         IS THIS THE FIRST ITERATION FOR THIS RUN.                      STG
  480 IF ( ITER .GT. RSTART + 1 )  GO TO 500                            STG
C         FIRST ITERATION.  IS SEGMENT FROM SEGID IN IDSEG ARRAY FROM    STG
C         SUBHST                                                         STG
      DO 490 I=1,NSEG                                                   STG
      IF ( ISEG .EQ. IDSEG(I) )  GO TO 500                              STG
  490 CONTINUE                                                          STG
C         SEGMENT IS NOT IN IDSEG.  PRINT WARNING AND DROP THIS SEGMENT  STG
      CALL ERRMES (3HSTG,3HSTG,2,0)                                     STG
      GO TO 300                                                         STG
C                                                                       STG
  500 PWKI= PWK                                                         STG
      PMKI= PWKI*DELTPM                                                 STG
C         TEST FOR ZERO ERROR CASE ( ISTG = 3 )                          STG
      IF ( ISTG .EQ. 3 )  GO TO 520                                     STG
C                                                                       STG
C         COMPUTE TRUE PW AND TRUE PM FOR THIS SEGMENT.                  STG
      SIGMA= PWK*CV2/100.0                                              STG
      CALL BETAD (SEED(1),PWK/100.0,SIGMA,PWKI,0,IERROR)                STG
      IARG(1)= IERROR                                                   STG
      IF(IERROR .NE. 0)CALL ERRMES(3HSTG,3HSTG,3,0)                             *NEW
      IF(IERROR .EQ. 3)PWKI = .01*PWK                                           *NEW
      PMMEAN= PWKI*DELTPM                                               STG     **-4
      SIGMA= PMMEAN*CV3                                                 STG
      CALL BETAD (SEED(1),PMMEAN,SIGMA,PMKI,0,IERROR)                   STG
      IARG(1)= IERROR                                                   STG
      IF(IERROR .NE. 0)CALL ERRMES(3HSTG,3HSTG,4,0)                             *NEW
      IF(IERROR .EQ. 3)PMKI = PMMEAN                                            *NEW
C                                                                       STG     **-4
      PWKI= 100.0*PWKI                                                  STG
      PMKI= 100.0*PMKI                                                  STG
C                                                                       STG
C         WRITE RECORD ON SEGMENT TRUTH FILE                             STG
  520 WRITE (SEGTRU)  CUNTRY,IREG,IZONE,ISTRAT,ISUBS,ISEG,ITRAIN,ITSPRL STG
     1  ,ISW,PWKI,PMKI                                                  STG
```

```
C          ADVANCE COUNTERS                                                 STG
      NSEGS= NSEGS + 1                                                      STG
      NREC(1)= NREC(1) + 1                                                  STG
      NSEGTR= NSEGTR + 1                                                    STG
      SUMPW= SUMPW + PWKI                                                   STG
      PW(NSEGS)= PWKI                                                       STG
      PM(NSEGS)= PMKI                                                       STG
      GO TO 300                                                             STG
C                                                                           STG
C          END OF A SUBSTRATUM.                                             STG
C          SET END-OF-DATA FLAG                                             STG
  600 ENDD= 1.0                                                             STG
C                                                                           STG
  610 IF ( PRINTF .EQ. 0 )  GO TO 650                                       STG
C          COMPUTE AVERAGE PW FOR THIS SUBSTRATA                            STG
      IF ( NSEGS .EQ. 0 )  GO TO 650                                        STG
      SNBR= NSEGS                                                           STG
      AVEPW= SUMPW/SNBR                                                     STG
C          COMPUTE ERROR IN SEGMENT PW                                      STG
      DO 630 I=1,NSEGS                                                      STG
      ERRPW(I)= PW(I) - PWK                                                 STG
  630 CONTINUE                                                              STG
C                                                                           STG
C          PRINT SEGMENT TRUTH REPORT DATA FOR THIS SUBSTRATA               STG
C                                                                           STG
      IF ( NLINE + NSEG+1 .LE. MXLINE )  GO TO 640                          STG
C          EJECT PAGE AND PRINT HEADERS BEFORE PRINTING SEGMENT TRUTH DATASTG
      CALL EJECT (5)                                                        STG
      WRITE (OUTP,1)                                                        STG
    1 FORMAT (/30X,39HS E G M E N T   T R U T H   R E P O R T//             STG
     1  11X,11HTRUE PW FOR,14X,11HTRUE PW FOR,6X,7HAVERAGE,9X,5HERROR       STG
     2  ,7X,11HTRUE PM FOR/32H SUBSTRATA  SUBSTRATA    SEGMENT,6X,          STG
     3  7HSEGMENT,10X,2HPW,12X,5HIN PW,9X,7HSEGMENT,5X,9HITERATION)         STG
  640 CALL PAGER (NSEG+1)                                                   STG
      WRITE (OUTP,2)  ISUBS2,PWK,IDSEG(1),PW(1),AVEPW,ERRPW(1),PM(1)        STG
     1  ,ITER                                                               STG
    2 FORMAT (/I7,F13.4,I10,4F15.4,I14)                                     STG
      IF ( NSEGS .EQ. 1 )  GO TO 650                                        STG
      WRITE (OUTP,3)  ( IDSEG(I),PW(I),ERRPW(I),PM(I),I=2,NSEGS )           STG
    3 FORMAT (I30,F15.4,15X,2F15.4)                                         STG
C          IF NSEGS .NE. NSEG, THEN PRINT WARNING THAT SEGMENT IDS ARE      STG
C          INCORRECT.                                                       STG
  650 IARG(1)= NSEGS                                                        STG
      IF ( NSEGS .NE. NSEG )  CALL ERRMES (3HSTG,3HSTG,5,0)                 STG
      NSEGS= 0                                                              STG
      SUMPW= 0.0                                                            STG
      IF ( ENDD .EQ. 0.0 )  GO TO 200                                       STG
C          END OF DATA ON SEGID AND/OR SUBHST                               STG
  900 NFILL= LSEGTR - 1                                                     STG
      WRITE (SEGTRU) ENDFIL, ( ZERO,I=1,NFILL )                             STG
      ENDFILE SEGTRU                                                        STG
      REWIND SEGTRU                                                         STG
      IF ( NSEGTR .EQ. 0 )  CALL ERRMES (3HSTG,3HSTG,6,0)                   STG
      RETURN                                                                STG
C                                                                           STG
      END                                                                   STG    **-2
```

```
@ ELT STGERR,1,760427, 39036  , 1
      SUBROUTINE STGERR (ICODE)                                          STGERR
C        PRINTS ERROR MESSAGES FOR SEGMENT TRUTH GENERATOR               STGERR
C                                                                        STGERR
C        ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                   ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG   ,ARG(10)                             ARGLST
      DIMENSION  IARG(10)                                                ARGLST
      EQUIVALENCE  ( IARG,ARG )                                          ARGLST
C                                                                        ARGLST
C        FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                   FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ           FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD          FILES
     4  ,INP    ,OUTP   ,TACQ   ,LTACQ ,CASDSF,LCASDS                    FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF   ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP   ,TACQ   ,CASDSF               FILES
C                                                                        FILES
C        DATA FOR SEGMENT TRUTH GENERATOR                                STGDTA
      COMMON  /STGDTA/                                                   STGDTA
     1   COUN  ,IREG  ,IZONE ,ISTRAT,ISUBS ,ISEG  ,ITRAIN,ITSPRL(6)      STGDTA
     2  ,SLAT  ,SLONG ,GRIDNO,ISW   ,COUN2 ,IREG2 ,IZONE2,ISTRA2         STGDTA
     3  ,ISUBS2,NSEG  ,IDSEG ,GRPNO ,HISTPW,AREA  ,PWK   ,NAGR           STGDTA
     4  ,NA    ,DELTPW,DELTPM,CV1   ,CV2   ,CV3   ,CV4   ,PWKI           STGDTA
     5  ,PMKI  ,AVEPW ,SUMPW ,SNBR  ,PMMEAN,PW    ,PM    ,ERRPW          STGDTA
     6  ,SIGMA ,ERRPWI                                                   STGDTA
      DIMENSION  IDSEG(150), PW(150), PM(150), ERRPW(150)                STGDTA
      INTEGER  GRIDNO,GRPNO                                              STGDTA
C                                                                        STGDTA
C                                                                        STGERR
      IMES= ICODE                                                        STGERR
      GO TO (100,200,300,400,500,600), IMES                              STGERR
C                                                                        STGERR
  100 WRITE (OUTP,1)  IREG,IREG2,IZONE,IZONE2,ISTRAT,ISTRA2,ISUBS,ISUBS2STGERR
    1 FORMAT (71H0THE SEGMENT ID FILE AND THE SUBSTRATA HISTORICAL FILE STGERR
     1ARE INCONSISTENT/13X,14HSEGID    SUBHST/7H REGION,I11,I9/5H ZONE,  STGERR
     2  I13,I9/7H STRATA,I11,I9/11H SUBSTRATA,I7,I9)                     STGERR
      GO TO 900                                                          STGERR
C                                                                        STGERR
  200 WRITE (OUTP,2)  ISEG,IREG,IZONE,ISTRAT,ISUBS                       STGERR
    2 FORMAT (9H0SEGMENT ,I5,41H  IS NOT IN IDSEG FROM SUBHST FOR REGIONSTGERR
     1 ,I4,8H , ZONE ,I4,10H , STRATA ,I4,13H , SUBSTRATA ,I4/           STGERR
     2  5X,23HSEGMENT WILL BE DROPPED)                                   STGERR
      GO TO 900                                                          STGERR
C                                                                        STGERR
  300 WRITE (OUTP,3)  IARG(1),PWK,SIGMA,PWKI                             STGERR
    3 FORMAT (40H0ERROR RETURN FROM BETAD ROUTINE.  IER= ,I2,8H , PWK=   STGERR
     1  E13.6,8H  SIGMA=,E13.6,7H  PWKI=,E13.6)
      GO TO 900                                                          STGERR
C                                                                        STGERR
  400 WRITE (OUTP,4)  IARG(1),PMMEAN,SIGMA,PMKI                          STGERR
    4 FORMAT(40H0ERROR RETURN FROM BETAD ROUTINE.  IER= ,I2,11H , PMMEANSTGERR
     1= ,E13.6,8H  SIGMA=,E13.6,7H  PMKI=,E13.6)
      GO TO 900                                                          STGERR
C                                                                        STGERR
  500 WRITE (OUTP,5)  IARG(1),NSEG                                       STGERR
    5 FORMAT (19H0WARNING... NSEGS= I4,29H  .NE.  NSEG (FROM SUBHST) =   STGERR
     1  I4/29H SEGMENT IDS MAY BE INCORRECT )                            STGERR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 900                                                         STGFRR
C                                                                       STGFRR
  600 WRITE (OUTP,6)                                                    STGFRR
    6 FORMAT (60H0WARNING... NO SEGMENTS PROCESSED BY SEGMENT TRUTH GENESTGERR
     1RATOR )                                                           STGFRR
C                                                                       STGFRR
  900 RETURN                                                            STGFRR
      END                                                               STGFRR
```

@ ELT SUMREP,1,760427, 39097 , 1

```
      SUBROUTINE SUMREP                                                 SUMREP
C        FLAGS AND COUNTERS FOR CAS SIMULATOR                           CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H      ,PPFLG ,NBW    ,IBW   ,WINDOW,IPD    ,IPP    ,PPDATE,NREGS  CASFLG
     2   ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NRYES ,NRSSH ,NRCAMS       CASFLG
     3   ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                      CASFLG
     4   ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13 CASFLG
     5   ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR        CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C        FILE DEFINITIONS AND RECORD LENGTHS                            FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2   ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO        FILES
     3   ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD        FILES
     4   ,THP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                     FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C        LEM CONTROL CARD INPUT DATA                                    LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)    ,ICASF ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2   ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3   ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4   ,ICSEYH,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5   ,RSEED7,ICSEST,ICSECU,ICSEYS,ICSECU,ICSECD                     LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1   ,RSEED6,RSEED7                                                 LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C        PAGE EJECT CONTROL PARAMETERS FOR LEM                          PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
C        STATISTICAL INFORMATION FOR LEM                                STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREL(7),NCASCR,NCASDP              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C        CAS CONTROL CARD INPUT DATA AND CONSTANTS                      CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF   ,PRDCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)   CASCM
     2   ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTEF,BWIND(4)     CASCM
     3   ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)          CASCM
      INTEGER  HH, TOPT, AUNITS,DISTEF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C        SUMMARY DATA FOR REPORTS                                       SUMDTA
      COMMON  /SUMDTA/                                                  SUMDTA
     1   CVALPT,CVLPTA,SDPER ,CVPEPT,CVEPTP,CSUMR(18,18)                SUMDTA
C                                                                       SUMDTA
C        CAS DATA SET 13 (COUNTRY DATA -- SECOND PASS)                  DSET13
      COMMON  /DSET13/                                                  DSET13
     1   HWAC  ,TWAC  ,EWAC  ,AERRC ,AVARC ,TPRODC,EPRODC,PRERRC,PRVARC DSET13
     2   ,TYC   ,EYC   ,YERRC ,M1C   ,M2C   ,CT1C  ,CT2C  ,CT3C  ,ANAVC DSET13
     3   ,ANPRVC,CLEWA ,CLEPRD,CLATEC,CLPTEC,CLATWC,CLPTWC              DSET13
      REAL  M1C , M2C                                                   DSET13
```

```
      DIMENSION  DSET13(25)                                             DSET13
      EQUIVALENCE  ( DSET13,HWAC ).                                     DSET13
C                                                                       DSET13
C                                                                       SUMREP
C  THIS SUBROUTINE PRINTS THE CAS COUNTRY SUMMARY REPORT                SUMREP
C                                                                       SUMREP
C       FIRST PAGE                                                      SUMREP
C            , HEADERS                                                  SUMREP
      CALL EJECT(13)                                                    SUMREP
      WRITE(OUTP,2000)                                                  SUMREP
 2000 FORMAT(/33X,45HC O U N T R Y    S U M M A R Y    R E P O R T//)   SUMREP
      WRITE(OUTP,3000) CUNTRY,NT                                        SUMREP
 3000 FORMAT(2X,9HCOUNTRY  ,A6,12H  ITERATION , I3)                     SUMREP
      WRITE(OUTP,4000)                                                  SUMREP
 4000 FORMAT(/12X,1H*,16X,8HA R E A ,16X,1H*,4X,10HY I E L D ,4X,       SUMREP
     1   1H*,8X,20HP R O D U C T I O N  )                               SUMREP
      INDX=AUNITS+1                                                     SUMREP
      WRITE(OUTP,5000) (APRUTS(I,INDX),I=1,4),(YPRUTS(I,INDX),I=1,3),   SUMREP
     1  (PPRUTS(I,INDX),I=1,5)                                          SUMREP
 5000 FORMAT(12X,1H*,10X,4A6,6X,1H*,3A6,1H*,6X,5A6)                     SUMREP
      WRITE(OUTP,6000)                                                  SUMREP
 6000 FORMAT(12X,1H*,40X,1H*,18X,1H*)                                   SUMREP
      WRITE(OUTP,7000)                                                  SUMREP
 7000 FORMAT(12X,1H*,13X,2(2HCV,8X),2HCV,5X,1H*,10X,6HST DEV,2X,        SUMREP
     1     1H*,12X,3(2HCV,8X))                                          SUMREP
      WRITE(OUTP,8000)                                                  SUMREP
 8000 FORMAT(1X,10HPREDICTION ,1X,1H*,1X,4HEST.,5X,7HANAL WA, 3X,       SUMREP
     1    8HAREA EST,4X,5HERROR,3X,1H*,2X,4HEST.,6X,3HPCT,3X,1H*,       SUMREP
     2   2X,4HEST.,4X,8HANAL PRD,3X,7HPRD EST,4X,5HERROR )              SUMREP
      WRITE(OUTP,9000)                                                  SUMREP
 9000 FORMAT(4X,5HPOINT,3X,1H*,2X,2HWA,5X,3(10H(PCT TRUE)),3H * ,       SUMREP
     1  5HYIELD,5X,5HERROR,2X,1H*,2X,3HPRD,4X,3(10H(PCT TRUE))/)        SUMREP
C                                                                       SUMREP
C   WRITE OUT PREDICTION POINTS FOR FIRST PAGE                          SUMREP
      IPP=0                                                             SUMREP
      DO 10 I=1,4                                                       SUMREP
      IF(PWIND(I).EQ.0) GO TO 10                                        SUMREP
      IPP=IPP+1                                                         SUMREP
      WRITE(OUTP,1500) I,(CSUMR(J,IPP),J=1,10)
 1500 FORMAT(6X,I1,3X,F10.1,5F10.2,F10.1,3F10.2)                        SUMREP
   10 CONTINUE                                                          SUMREP
      IF(NPDATE.EQ.0) GO TO 30                                          SUMREP
      DO 20 I=1,NPDATE                                                  SUMREP
      IPP=IPP+1                                                         SUMREP
      WRITE(OUTP,2500) IPRD(2,I),IPRD(3,I),IPRD(1,I),                   SUMREP
     1                 (CSUMR(J,IPP),J=1,10)
 2500 FORMAT(2X,2(I2,1H/),I2,F10.1,5F10.2,F10.1,3F10.2)                 SUMREP
   20 CONTINUE                                                          SUMREP
   30 CONTINUE                                                          SUMREP
      CALL PAGER(IPP)                                                   SUMREP
C                                                                       SUMREP
C   WRITE OUT SUMMARY INFORMATION ON FIRST PAGE                         SUMREP
      CALL PAGER(6)                                                     SUMREP
      HWAC= HWAC*AREACF                                                 SUMREP
      TWAC= TWAC*AREACF                                                 SUMREP
      TYC = TYC *YCF                                                    SUMREP
      TPRODC= TPRODC*PRDCF                                              SUMREP
      WRITE(OUTP,3500) HWAC,(APRUTS(I,INDX),I=1,4)                      SUMREP
 3500 FORMAT(///2X,15HHISTORICAL AREA,3X,F10.2,3X,4A6)                  SUMREP
      WRITE(OUTP,4500) TWAC,(APRUTS(I,INDX),I=1,4)                      SUMREP
```

```
  4500 FORMAT(2X,9HTRUE AREA,9X,F10.2,3X,4A6)                            SUMREP
       WRITE(OUTP,5500) TYC,(YPRUTS(I,INDX),I=1,3)                       SUMREP
  5500 FORMAT(2X,10HTRUE YIELD,8X,F10.2,3X,3A6)                          SUMREP
       WRITE(OUTP,6500) TPRODC,(PPRUTS(I,INDX),I=1,5)                    SUMREP
  6500 FORMAT(2X,15HTRUE PRODUCTION, 3X,F10.2,3X,5A6)                    SUMREP
C                                                                        SUMREP
C   SECOND PAGE                                                          SUMREP
C          HEADERS                                                       SUMREP
       CALL EJECT(11)                                                    SUMREP
       WRITE(OUTP,2000)                                                  SUMREP
       WRITE(OUTP,3000) CUNTRY,NT                                        SUMREP
       WRITE(OUTP,7500)                                                  SUMREP
  7500 FORMAT(//30X,22HAREA CONFIDENCE LEVELS,24X,                       SUMREP
      1   28HPRODUCTION CONFIDENCE LEVELS )                              SUMREP
       WRITE(OUTP,8500)                                                  SUMREP
  8500 FORMAT(1X,10HPREDICTION )                                         SUMREP
       WRITE(OUTP,9500)                                                  SUMREP
  9500 FORMAT(4X,5HPOINT,12X,2(10HTRUE/ERROR,2X,7HEST/EST,2X,            SUMREP
      1   8HTRUE/EST,3X,7HTRUE/RC,11X)/)                                 SUMREP
C                                                                        SUMREP
C   WRITE OUT PREDICTION POINTS FOR SECOND PAGE                          SUMREP
       IPP=0                                                             SUMREP
       DO 40 I=1,4                                                       SUMREP
       IF(BWIND(I).EQ.0) GO TO 40                                        SUMREP
       IPP=IPP+1                                                         SUMREP
       WRITE(OUTP,1100) I,(CSUMR(J,IPP),J=11,18)
  1100 FORMAT(6X,I1,13X,2(4F10.3,10X))                                   SUMREP
    40 CONTINUE                                                          SUMREP
       IF(NPDATE.EQ.0) GO TO 50                                          SUMREP
       DO 60 I=1,NPDATE                                                  SUMREP
       IPP=IPP+1                                                         SUMREP
       WRITE(OUTP,2100) IPRD(2,I),IPRD(3,I),IPRD(1,I),                   SUMREP
      1    (CSUMR(J,IPP),J=11,18)
  2100 FORMAT(2X,2(I2,1H/),I2,10X,2(4F10.3,10X))                         SUMREP
    60 CONTINUE                                                          SUMREP
    50 CONTINUE                                                          SUMREP
       CALL PAGER(IPP)                                                   SUMREP
       CALL PAGEP(3)                                                     SUMREP
       WRITE(OUTP,3100)                                                  SUMREP
  3100 FORMAT(//10X,                                                     SUMREP
      1 53HTRUE/ERROR AREA AND PRODUCTION CONFIDENCE LEVELS ARE  ,       SUMREP
      1 35HCALCULATED ONLY FOR FINAL ITERATION )                         SUMREP
C        SET NLINE TO EJECT PAGE AFTER COUNTRY SUMMARY REPORT BEFORE     SUMREP
C        PRINTING ANY MORE OUTPUT                                        SUMREP
       NLINE= MXLINE + 1                                                 SUMREP
   900 RETURN                                                            SUMREP
       END                                                               SUMREP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE TSAVE(ISEG,IOPT,IBAD)                                  TSAVE
C                                                                       TSAVE
C     THIS SUBROUTINE HANDLES THE I/O FOR THE SCRATCH RA FILE TACQ FOR CAMTSAVE
C                                                                       TSAVE
      COMMON/TRAINS/  COUN7,IREG7,IZONE7,ISTRA7,ISUB7,ISEG7,            TRAINS
     1   ITWIN(4,25),TTTOT,TMM(3,4,25),TBB(3,4,25),TVV(3,4,25),         TRAINS
     1   TPTRUE,TIZULU(4),TPEST(4),TPERR(4),TERTOT(3),TM(3),TV(3),TB(3) TRAINS
      INTEGER TIZULU                                                    TRAINS
      DIMENSION ITRAIN(129)                                             TRAINS
      EQUIVALENCE(ITRAIN,COUN7)                                         TRAINS
      COMMON/INDX/ INDEX(    1),IPOINT(2001),IPNT2(2001),IPEND,IPIN     INDX
C       FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF   ,LCASF ,YESOUT,LYESU         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP    ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                     FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C          ARGUMENT LIST FOR ERROR PROCESSING                           ARGLST
      COMMON  /ARGLST/                                                  ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C                                                                       TSAVE
C  OPEN FILE                                                            TSAVE
      IBAD=0                                                            TSAVE
      IF(IOPT.NE.0) GO TO 10                                            TSAVE
      L=50                                                              TSAVE
      DEFINE FILE 16(50,1020,U,IDUM)                                             *NEW
      RETURN                                                            TSAVE    **-1
C                                                                       TSAVE
C   CLOSE FILE                                                          TSAVE
   10 CONTINUE                                                          TSAVE
      IF(IOPT.NE.-1) GO TO 20                                           TSAVE
      CALL RANACF(TACQ,0,0,0,0,0,-1)                                    TSAVE
      RETURN                                                            TSAVE
C                                                                       TSAVE
C   WRITE TO FILE, SEQUENTIALLY                                         TSAVE
   20 CONTINUE                                                          TSAVE
      IF(IOPT.NE.2) GO TO 25                                            TSAVE
      IPEND=IPEND+1                                                     TSAVE
      IF(IPEND.LE.2000) GO TO 21                                        TSAVE
      NARG=0                                                            TSAVE
      CALL ERRMES(4HCAMS,5HTSAVE,4,1)                                   TSAVE
      RETURN                                                            TSAVE
   21 CONTINUE                                                          TSAVE
      IPOINT(IPEND)=ISEG                                                TSAVE
      IPNT2(IPEND)=IPEND                                                TSAVE
      CALL RANACF(TACQ,IPEND,ITRAIN,1020,INDEX,1,2)                     TSAVE
      RETURN                                                            TSAVE
C                                                                       TSAVE
C   FINISHED WRITES - SORT IPOINT,IPNT2                                 TSAVE
   25 CONTINUE                                                          TSAVE
      IF(IOPT.NE.3) GO TO 30                                            TSAVE
```

```
      IPOINT(IPEND+1)=9999999                                          TSAVE
      CALL SORTAG(IPOINT,1,IPEND,IPNT2)                                TSAVE
      CALL RANACF(TACQ,0,0,0,0,0,-1)                                   TSAVE
      RETURN                                                           TSAVE
C                                                                      TSAVE
C   READ FROM FILE                                                     TSAVE
C                                                                      TSAVE
C   FIND INDEX TO INDEX, BINARY SEARCH                                 TSAVE
   30 CONTINUE                                                         TSAVE
      IL=1                                                             TSAVE
      IH=IPEND                                                         TSAVE
   32 CONTINUE                                                         TSAVE
      ILOOK=(IH+IL)/2                                                  TSAVE
      IF(IPOINT(ILOOK).EQ.ISEG) GO TO 35                               TSAVE
      IF(IPOINT(ILOOK).GT.ISEG) IH=ILOOK-1                             TSAVE
      IF(IPOINT(ILOOK)  .LT.ISEG) IL=ILOOK+1                           TSAVE
      IF(IH.GE.IL) GO TO 32                                            TSAVE
      IBAD=1                                                           TSAVE
      RETURN                                                           TSAVE
C                                                                      TSAVE
C   FOUND CORRECT INDEX                                                TSAVE
   35 CONTINUE                                                         TSAVE
C                                                                      TSAVE
C   READ IN RECORD IF NOT ALREADY READ                                 TSAVE
      IPIN=IPNT2(ILOOK)                                                TSAVE
      CALL RANACF(TACQ,IPIN,ITRAIN,1020,INDEX,1,1)                     TSAVE
      RETURN                                                           TSAVE
      END                                                              TSAVE
```

```
FOR,IS TSUB
      SUBROUTINE TSUB                                                   TSUB
C        COMPUTES THE QUANTITY T (SECOND TERM OF PPS AREA VARIANCE EQN.)TSUB
C        T IS GIVEN BY EQ. 39 IN CAS PROBLEM DESCRIPTION.               TSUB
C                                                                       TSUB
C        DATA BLOCK FOR CAS CUMULATIVE FILE                             CASCUM
C        CAS DATA SETS 14, 15, 16, AND 17                               CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),    BUFFR(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C        DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)             CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFR )                                     CASDSB
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1  , ( PIK,CASDSB(121) )                                           CASDSB
C                                                                       CASDSB
C        CAS DATA SETS 4, 5, AND 6 (AT STRATA LEVEL)                    DSET4
      COMMON  /DSET4 /                                                  DSET4
     1   STRATA,TWAS1 ,HWAS1 ,EWAS1 ,XM1JS ,XCT1S ,ANVS1                JULY76
     2  ,TWAS2 ,HWAS2 ,EWAS2 ,XM2JS ,XCT2S ,ANVS2 ,T                    JULY76
     3 ,TWAS3,HWAS3,XCT3S
     4  ,XYS    ,XESTYS,EVYRS ,P2IDPK,V1V2S ,VARS   ,ANVARS             JULY76
     5 ,FILL4(57)
      INTEGER  STRATA                                                   JULY76
      DIMENSION  DSET4(24), DSET5(7), DSET6(3)                          JULY76
      EQUIVALENCE  ( DSET4,STRATA ), ( DSET5,TWAS2 ), ( DSET6,TWAS3 )   DSET4
C                                                                       DSET4
C                                                                       TSUB
      CON = XM2JS/HWAS2                                                 TSUB
      NS2 = XCT2S                                                       TSUB
C        COMPUTE ALL PI(K), THE SUM OF PI(K)**2 OVER ALL SUBSTRATA,     TSUB
C        AND THE SUM OF PI(K)**3 OVER ALL SUBSTRATA.                    TSUB
      SUM2= 0.0                                                         TSUB
```

```
      SUM3= 0.0                                                         TSUB
      DO 110 K=1,NS2                                                    TSUB
      PIK(K)= CON*HWA2K(K)                                              TSUB
      SUM2= SUM2 + PIK(K)**2                                            TSUB
      SUM3= SUM3 + PIK(K)**3                                            TSUB
  110 CONTINUE                                                          TSUB
C                                                                       TSUB
C        COMPUTE CONSTANTS WHICH DEPEND ONLY UPON M2JS AND HWAS2        TSUB
C        INDEPENDENT OF SUBSTRATA                                       TSUB
      CON1= (XM2JS-1.0)/XM2JS                                           TSUB
      CON2= CON1/XM2JS                                                  TSUB
      CON3 = 2.0*CON2/XM2JS                                             TSUB
      CON3S= CON2*SUM2/XM2JS                                            TSUB
      CON4S= 3.0*CON3S/XM2JS                                            TSUB
      CON4S3= CON3*SUM3/XM2JS                                           TSUB
      CON5S2= CON4S*SUM2/XM2JS                                          TSUB
C                                                                       TSUB
      NS2M1= NS2 - 1                                                    TSUB
      DO 210 K=1,NS2M1                                                  TSUB
      WAKPIK= WAKNEY(K)/PIK(K)                                          TSUB
      PIK2= PIK(K)**2                                                   TSUB
      PIK3= PIK2*PIK(K)                                                 TSUB
C                                                                       TSUB
      KP1= K + 1                                                        TSUB
      DO 210 KP=KP1,NS2                                                 TSUB
      PIKPKP= PIK(K)*PIK(KP)                                            TSUB
      TERM2= PIK2*PIK(KP) + PIK(K)*PIK(KP)**2                           TSUB
C                                                                       TSUB
      PIKPP= CON1*PIKPKP + CON2*TERM2 - CON3S*PIKPKP                    TSUB
     1   + CON3*( PIK3*PIK(KP) + PIK(K)*PIK(KP)**3 + PIK2*PIK(KP)**2 )  TSUB
     2   - CON4S*TERM2 + CON5S2*PIKPKP - CON4S3*PIKPKP                  TSUB
C                                                                       TSUB
      T= T + ( PIKPKP - PIKPP )* ( WAKPIK - WAKNEY(KP)/ PIK(KP) )**2    TSUB
  210 CONTINUE                                                          TSUB
C                                                                       TSUB
  900 RETURN                                                            TSUB
      END                                                               TSUB
```

@ ELT WRAPUP,1,760430, 50107 , 1

```
      SUBROUTINE WRAPUP                                                 WRAPUP
C       WRITES HEADER RECORDS ON CAS CUMULATIVE FILE AND CAS DISTR. FILEWRAPUP
C       CLOSES RANDOM ACCESS FILES (CASCUM AND CASDIS)                  WRAPUP
C       PRINTS STATUS INFORMATION AT END OF RUN.                        WRAPUP
C                                                                       WRAPUP
C         ARGUMENT LIST FOR ERROR PROCESSING                            ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C         CAS CONTROL CARD INPUT DATA AND CONSTANTS                     CASCM
      COMMON  /CASCM /                                                  CASCM
     1   AREACF,YCF   ,PROCF ,APRUTS(4,2)  ,PPRUTS(5,2)  ,YPRUTS(3,2)   CASCM
     2  ,AREAPS,S2MAX ,NHISTY,HH    ,TOPT  ,AUNITS,DISTFF,BWIND(4)      CASCM
     3  ,WPRIOR(4)    ,APREP ,IPRD(3,14)   ,NPDATE,PRDATE(14)           CASCM
      INTEGER  HH, TOPT, AUNITS,DISTFF,BWIND,WPRIOR,APREP,PRDATE        CASCM
C                                                                       CASCM
C         DATA BLOCK FOR CAS CUMULATIVE FILE                            CASCUM
C         CAS DATA SETS 14, 15, 16, AND 17                              CASCUM
      COMMON  /CASCUM/                                                  CASCUM
     1   CASCUM(32),   BUFFER(504)                                      CASCUM
      DIMENSION  ICASC(32), DSET14(22), DSET15(22), DSET16(22)          CASCUM
     1  ,DSET17(28)                                                     CASCUM
      EQUIVALENCE  ( ICASC,CASCUM )                                     CASCUM
      EQUIVALENCE  ( DSET14,DSET15,DSET16,DSET17,CASCUM(5) )            CASCUM
     1  , ( SQAERS,SQAERZ,SQAERR,SQAERC,CASCUM(24) )                    CASCUM
     2  , ( SQPERS,SQPERZ,SQPERR,SQPERC,CASCUM(25) )                    CASCUM
     3  , ( SQYERS,SQYERZ,SQYERR,SQYERC,CASCUM(26) )                    CASCUM
C                                                                       CASCUM
C         DATA BLOCK FOR CAS DISTRIBUTION FILE (DATA SET 19)            CASDSB
      DIMENSION  CASDSB(303)                                            CASDSB
      EQUIVALENCE  ( CASDSB,BUFFER )                                    CASDSB
      DIMENSION ICASD(303), HWA2K(60), WAKNEY(60), PIK(60)              CASDSB
      EQUIVALENCE  ( ICASD,HWA2K,CASDSB ), ( WAKNEY,CASDSB(61) )        CASDSB
     1  , ( PIK,CASDSB(121) )                                           CASDSB
C                                                                       CASDSB
C         FLAGS AND COUNTERS FOR CAS SIMULATOR                          CASFLG
      COMMON  /CASFLG/                                                  CASFLG
     1   H     ,PPFLG ,NBW   ,IBW   ,WINDOW,IPD   ,IPP   ,PPDATE,NREGS  CASFLG
     2  ,NZTOT ,NSTRAT,NYESSK,NSSHSK,NCAMSK,NPYES ,NRSSH ,NRCAMS         CASFLG
     3  ,ENDC  ,ENDREG,ENDZON,IRSTR ,IRZONE,IRREG                       CASFLG
     4  ,LDS1  ,LDS4  ,LDS7  ,LDS8  ,LDS9  ,LDS10 ,LDS11 ,LDS12 ,LDS13  CASFLG
     5  ,LDS14 ,LDS15 ,LDS16 ,LDS17 ,LRCOUN,LRREG ,LRZONE,LRSTR         CASFLG
      INTEGER  PPFLG , WINDOW , PPDATE                                  CASFLG
C                                                                       CASFLG
C         CONTROL PARAMETERS FOR LEM PROGRAM                            CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C         FILE DEFINITIONS AND RECORD LENGTHS                           FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACOUTS,LACO          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGIPD,LSEGIP,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACO  ,LTACO ,CASDSF,LCASDS                      FILES
```

```
      INTEGER  SEGID ,CROPW ,SUBHSI,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                 FILES
C                                                                        FILES
C        INDEX RECORD FOR CAS CUMULATIVE FILE (CASF)                     IXCASF
      COMMON  /IXCASF/                                                   IXCASF
     1   IXCASF(  1)  ,LIXCAS                                            IXCASF
C                                                                        IXCASF
C        DATA BLOCK FOR CAS DISTRIBUTION FILE                            IXDISF
      COMMON  /IXDISF/                                                   IXDISF
     1   IXDISF(  1), LIXDIS                                             IXDISF
C        NOTE... 506 ONLY ALLOWS UP TO 8 PREDICTION POINTS INCLUDING     IXDISF
C        BIOWINDOWS ( 506 = 1 + 1 + 8*63, INDEX + HEADER + 8 PRED. PTS.)IXDISF
C                                                                        IXDISF
C        LEM CONTROL CARD INPUT DATA                                     LEMCM
      COMMON  /LEMCM /                                                   LEMCM
     1   TITLE(10)     ,ICASF ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG  ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC   LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECW,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESF,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECU,ICSEYS,ICSECU,ICSECP                       LEMCM
      DIMENSION  RSEED(7)                                                LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5        LEMCM
     1  ,RSEED6,RSEED7                                                   LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                      LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                          LEMCM
C                                                                        LEMCM
C        STATISTICAL INFORMATION FOR LEM                                 STATS
      COMMON  /STATS /                                                   STATS
     1   ITER  ,NSEGIR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR               STATS
      EQUIVALENCE  ( NT,ITER )                                           STATS
C                                                                        STATS
C                                                                        WRAPUP
C        WRITE HEADER RECORD ON CAS CUMULATIVE FILE                      WRAPUP
      CASCUM(1)= 6HCASCUM                                                WRAPUP
      ICASC(2) = ICASE                                                   WRAPUP
      CASCUM(3)= CUNTRY                                                  WRAPUP
      ICASC(4)= NT                                                       WRAPUP
      ICASC(5) = NREGS                                                   WRAPUP
      ICASC(6) = NZTOT                                                   WRAPUP
      ICASC(7) = NSTRAT                                                  WRAPUP
C                                                                        WRAPUP
C        STORE BIOWINDOW NUMBERS IN WORDS 10-13 OF HEADER RECORD         WRAPUP
      NBW= 0                                                             WRAPUP
      DO 110 I=1,4                                                       WRAPUP
      ICASC(I+9)= 0                                                      WRAPUP
      IF ( BWIND(I) .EQ. 0 )  GO TO 110                                  WRAPUP
      NBW= NBW + 1                                                       WRAPUP
      ICASC(NBW+9)= I                                                    WRAPUP
  110 CONTINUE                                                           WRAPUP
C                                                                        WRAPUP
C        STORE NBW AND NPDATE IN WORDS 8 AND 9 OF HEADER RECORD          WRAPUP
      ICASC(8)= NBW                                                      WRAPUP
      ICASC(9)= NPDATE                                                   WRAPUP
C        STORE ZULU PREDICTION DATES IN WORDS 14-27 OF HEADER RECORD     WRAPUP
      DO 120 I=1,14                                                      WRAPUP
      ICASC(I+13)= PRDATE(I)                                             WRAPUP
  120 CONTINUE                                                           WRAPUP
C                                                                        WRAPUP
C        FILL IN REST OF HEADER RECORD WITH ZEROS                        WRAPUP
      DO 130 I=28,LCASF                                                  WRAPUP
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      ICASC(1)= 0                                                       WRAPUP
  130 CONTINUE                                                          WRAPUP
C                                                                       WRAPUP
      CALL RANACF ( CASF,1,CASCUM,LCASF,IXCASF,LIXCAS,2)                WRAPUP
C        CLOSE CAS CUMULATIVE FILE                                      WRAPUP
      CALL RANACF (CASF,0,0,0,0,0,-1)                                   WRAPUP
C                                                                       WRAPUP
      IF ( DISTFF .EQ. 0 )  GO TO 800                                   WRAPUP
C                                                                       WRAPUP
C        WRITE HEADER RECORD ONTO CAS DISTRIBUTION FILE.                WRAPUP
C        NOTE THAT CASCUM ARRAY IS BEING USED INSTEAD OF THE CASDIS     WRAPUP
C        ARRAY.                                                         WRAPUP
      CASCUM(1)= 6HCASDIS                                               WRAPUP
      ICASC(7)= 0                                                       WRAPUP
      CALL RANACF (CASDIS,1,CASCUM,LCASD,IXDISF,LIXDIS,2)               WRAPUP
C        CLOSE CAS DISTRIBUTION FILE                                    WRAPUP
      CALL RANACF (CASDIS,0,0,0,0,0,-1)                                 WRAPUP
C                                                                       WRAPUP
C        PRINT PROGRAM STATUS INFORMATION                               WRAPUP
  800 NREC(3)= NRSSH - NSSHSK + 1                                       WRAPUP
      N1     = NRCAMS - NCAMSK + 1                                      WRAPUP
      N2     = NRYES  - NYESSK + 1                                      WRAPUP
      NCASCR = NREGS + NZTOT + NSTRAT + 1                               WRAPUP
      NCASDR = (NREGS + NZTOT + 3)*IPP                                  WRAPUP
C                                                                       WRAPUP
      CALL EJECT (33)                                                   WRAPUP
      WRITE (OUTP,1)  NSTART,NT,NREC,N1,N2,NSEGTR,NCAMSR,NYESR,NCASCR   WRAPUP
     1  ,NCASDR,SEED,NPERRS                                             WRAPUP
C                                                                       WRAPUP
    1 FORMAT (/30X,51HP R O G R A M   S T A T U S   I N F O R M A T I O WRAPUP
     1N //5X,I5,5X,29HINITIAL MONTE CARLO ITERATION                     WRAPUP
     2   /5X,I5,5X,29HFINAL   MONTE CARLO ITERATION                     WRAPUP
     3  //5X,I5,5X,33HRECORDS READ FROM SEGMENT ID FILE                 WRAPUP
     4   /5X,I5,5X,34HRECORDS READ FROM CROP WINDOW FILE                WRAPUP
     5   /5X,I5,5X,43HRECORDS READ FROM SUBSTRATA HISTORICAL FILE       WRAPUP
     6   /5X,I5,5X,33HRECORDS READ FROM CAMS ERROR FILE                 WRAPUP
     7   /5X,I5,5X,38HRECORDS READ FROM YES ERROR MODEL FILE            WRAPUP
     8   /5X,I5,5X,42HRECORDS READ FROM SIGNATURE EXTENSION FILE        WRAPUP
     9   /5X,I5,5X,39HRECORDS READ FROM DATA ACQUISITION FILE           WRAPUP
     A  //5X,I5,5X,34HRECORDS READ FROM CAMS OUTPUT FILE                WRAPUP
     B   /5X,I5,5X,38HRECORDS READ FROM YES ERROR MODEL FILE            WRAPUP
     C   /5X,I5,5X,39HRECORDS WRITTEN ONTO SEGMENT TRUTH FILE           WRAPUP
     D   /5X,I5,5X,37HRECORDS WRITTEN ONTO CAMS OUTPUT FILE             WRAPUP
     E   /5X,I5,5X,36HRECORDS WRITTEN ONTO YES OUTPUT FILE              WRAPUP
     F   /5X,I5,5X,40HRECORDS WRITTEN ONTO CAS CUMULATIVE FILE          WRAPUP
     G   /5X,I5,5X,42HRECORDS WRITTEN ONTO CAS DISTRIBUTION FILE        WRAPUP
     H  //8X,22HRANDOM NUMBER SEEDS = D19.12/6(30X,D19.12/)             WRAPUP
     I//5X,I5,5X,42H          ERRORS DETECTED DURING EXECUTION)                 *NEW
  900 RETURN                                                                    **-1
      END
```

@ ELT YES,1,760428, 74670 , 1

```
      SUBROUTINE YES                                                    YES
C                                                                       YES
C   THIS SUBROUTINE CALCULATES THE ESTIMATED YIELD FROM THE TRUE YIELD  YES
C       AND A RANDOM NUMBER FROM A BETA DISTRIBUTION, A BIAS AND A      YES
C       STANDARD DEVIATION.   IT NEEDS INPUT FILE YESERR AND PRODUCES   YES
C       OUTPUT FILE YESOUT, ALSO AN OPTIONAL REPORT.                    YES
C                                                                       YES
C       LEM CONTROL CARD INPUT DATA                                     LEMCM
      COMMON  /LEMCM /                                                  LEMCM
     1   TITLE(10)     ,ICASE ,CUNTRY,NTRIAL,RSTART,IPRINT,STARTR,STARTZ LEMCM
     2  ,ENDR  ,ENDZ  ,ISTG   ,ICAMS ,IYES  ,IACQ  ,ICLASS,ISEXT ,ISCC  LEMCM
     3  ,ICAS2 ,ICAS3 ,IPRCAM,IPRYES,IPRCAS,ICSESG,ICSECR,ICSESH,ICSECE LEMCM
     4  ,ICSEYM,ICSESE,ICSEAC,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5,RSEED6 LEMCM
     5  ,RSEED7,ICSEST,ICSECO,ICSEYS,ICSECU,ICSECD                      LEMCM
      DIMENSION  RSEED(7)                                               LEMCM
      DOUBLE PRECISION  RSEED ,RSEED1,RSEED2,RSEED3,RSEED4,RSEED5       LEMCM
     1  ,RSEED6,RSEED7                                                  LEMCM
      EQUIVALENCE  ( RSEED,RSEED1 )                                     LEMCM
      INTEGER  RSTART,STARTR,STARTZ,ENDR  ,ENDZ                         LEMCM
C                                                                       LEMCM
C       CONTROL PARAMETERS FOR LEM PROGRAM                              CNTRL
      COMMON  /CNTRL /                                                  CNTRL
     1   PRINTF,NSTART,SEED(7)                                          CNTRL
      INTEGER  PRINTF                                                   CNTRL
      DOUBLE PRECISION  SEED                                            CNTRL
C                                                                       CNTRL
C       ARGUMENT LIST FOR ERROR PROCESSING                              ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C       FILE DEFINITIONS AND RECORD LENGTHS                             FILES
      COMMON  /FILES /                                                  FILES
     1   SEGID ,LSEGID,CROPW ,LCROPW,SUBHST,LSUBH ,ACQUIS,LACQ          FILES
     2  ,CAMSF ,LCAMSF,CAMERR,LCAMER,CASF  ,LCASF ,YESOUT,LYESO         FILES
     3  ,SIGEXT,LSIGEX,YESERR,LYESER,SEGTRU,LSEGTR,CASDIS,LCASD         FILES
     4  ,INP   ,OUTP  ,TACQ  ,LTACQ ,CASDSF,LCASDS                      FILES
      INTEGER  SEGID ,CROPW ,SUBHST,ACQUIS,CAMSF ,CAMERR,CASF  ,YESOUT  FILES
     1 ,SIGEXT,YESERR,SEGTRU,CASDIS,OUTP  ,TACQ  ,CASDSF                FILES
C                                                                       FILES
C       STATISTICAL INFORMATION FOR LEM                                 STATS
      COMMON  /STATS /                                                  STATS
     1   ITER  ,NSEGTR,NCAMSR,NYESR ,NREC(7),NCASCR,NCASDR              STATS
      EQUIVALENCE  ( NT,ITER )                                          STATS
C                                                                       STATS
C       PAGE EJECT CONTROL PARAMETERS FOR LEM                           PAGECM
      COMMON  /PAGECM/                                                  PAGECM
     1   NPAGE ,NLINE ,MXLINE,NSTTL ,SUBTTL(10)                         PAGECM
C                                                                       PAGECM
      COMMON/YESIN/  COUN,IREG,IZONE,ISTRAT,YTRUE,IZULU(6),BIAS(6),     YES
     1   SD(6)                                                          YES
      COMMON/YESOT/   CID,IREGID,IZONID,ISTRID,YSTR,IZPROD(6),YSCT(6),  YES
     1  YSYCT(6)                                                        YES
      DIMENSION YNAME(2),IOUT(3)                                        YES
      DATA IFILL /0/                                                    YES
      DATA YNAME(1)/3HYES/,YNAME(2)/1H /                                YES
      DATA ZZZZ/4HZZZZ/                                                 YES
```

```
      DATA INEW /0/                                                     YES
      INEW = 0                                                                  *NEW
      REWIND YESERR                                                     YES
      REWIND YESOUT                                                     YES
      NYESR=0                                                           YES
      NREC(5)=1                                                         YES
      NARG=0                                                            YES
      NSTTL=0                                                           YES
C                                                                       YES
C   SKIP HEADER                                                         YES
      READ(YESERR)                                                      YES
C                                                                       YES
C   SKIP TO BEGINNING ZONE                                              YES
   10 READ(YESERR) COUN,IREG,IZONE,ISTRAT,YTRUE,                        YES
     1  (IZOU(I),BIAS(I),SD(I),I=1,6)                                   YES
      IF(COUN.NE.ZZZZ) GO TO 20                                         YES
C                                                                       YES
C   IF CANNOT FIND START ZONE, REPORT ERROR                             YES
      CALL ERRMES(3HYES,3HYES,1,1)                                      YES
      RETURN                                                            YES
   20 CONTINUE                                                          YES
C                                                                       YES
C   IF FOUND START RECORD - START PROCESSING RECORDS                    YES
      IF((IREG.NE.STARTR.OR.IZONE.NE.STARTZ).AND.STARTR.NE.0) GO TO 10  YES
      ITEMP=LYESO-3                                                     YES
C   WRITE HEADER TO OUTPUT FILE                                         YES
      WRITE(YESOUT)YNAME(1),YNAME(2),ICASE,(IFILL,I=1,ITEMP)            YES
   22 CONTINUE                                                          YES
      LID=COUN                                                          YES
      IREGID=IREG                                                       YES
      IZONID=IZONE                                                      YES
      ISTRID=ISTRAT                                                     YES
      YSTR=YTRUE                                                        YES
C                                                                       YES
C   WRITE HEADING ON OUTPUT REPORT IF OPTION ON                         YES
      IF(PRINTF.EQ.0) GO TO 28                                          YES
      INEW=INEW+1                                                       YES
      IF(INEW.GE.4) INEW=1                                              YES
      IF(INEW.GT.1) GO TO 27                                            YES
      CALL EJECT(2)                                                     YES
      WRITE(OUTP,1020)                                                  YES
      WRITE(OUTP,1000)ITER                                              YES
 1000 FORMAT(32X,48HYES YIELD ESTIMATE DATA REPORT   -ITERATION NO.  ,  YES
     1  I5)                                                             YES
   27 CONTINUE                                                          YES
      CALL PAGER(6)                                                     YES
      WRITE(OUTP,1020)                                                  YES
      WRITE(OUTP,1020)                                                  YES
      WRITE(OUTP,1010)COUN,IREG,IZONE,ISTRAT                            YES
 1010 FORMAT( 2X,8HCOUNTRY ,A4,8H REGION ,I2,6H ZONE ,I3,9H STRATUM ,I3)YES
      WRITE(OUTP,1020)                                                  YES
 1020 FORMAT(1X)                                                        YES
      WRITE(OUTP,1030)                                                  YES
 1030 FORMAT(6X,14HPREDICT.DATE  ,12HTRUE YIELD  ,2X,13HESTIM.YIELD   , YES
     1 1X,  10HPERCENT   ,15HSTANDARD DEV.  )                           YES
      WRITE(OUTP,1040)                                                  YES
 1040 FORMAT(6X,8HMO/DY/YR ,6X ,14HQUIN./HECTAR  ,14HQUIN./HECTAR  ,    YES
     1  10HERROR     ,14HQUIN./HECTAR  )                                YES
   28 CONTINUE                                                          YES
C   INITIALIZE OUTPUT RECORD TO ZEROS                                   YES
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      DO 35 J=1,6                                                          YES
      IZPRDD(J)=0                                                          YES
      YSCI(J)=0.                                                           YES
      VSYCI(J)=0.                                                          YES
   35 CONTINUE                                                             YES
C                                                                          YES
C     OUTPUT RECORD FOR EACH PREDICTION POINT, COMPUTING ESTIM. YIELD      YES
      DO 40 J=1,6                                                          YES
      IF(IZULU(J).EQ.0) GO TO 50                                           YES
      IZPRDD(J)=IZULU(J)                                                   YES
      VSYCI(J)=SD(J)                                                       YES
      YSCI(J)=YTRUE                                                        YES
C                                                                          YES
C     CHECK FOR NO ERROR ESTIM. OPTION                                     YES
      IF(IYES.EQ.3) GO TO 30                                               YES
      CALL BETAD(SEED(5),0.,0.,RN,1,IER)                                   YES
      YSCI(J)=YSTR +BIAS(J)+RN*SD(J)                                       YES
      IF(YSCI(J).LT.0.0) YSCI(J)=0.0                                       YES
   30 CONTINUE                                                             YES
C                                                                          YES
C     IF REPORT OPTION ON, PRINT LINE ON REPORT                            YES
      IF(PRINTF.EQ.0) GO TO 40                                             YES
      CALL FZULU(IZULU(J),IOUT)                                            YES
      ER=0.000001                                                          YES
      IF(YSTR.GT.0.0)ER=0.0                                                YES
      PERCNT=ABS(YSCI(J)-YSTR)/(YSTR+ER)*100.                              YES
      CALL PACER(1)                                                        YES
      WRITE(OUTP,1060)IOUT(2),IOUT(3),IOUT(1),YSTR,YSCI(J),PERCNT,SD(J)    YES
 1060 FORMAT(6X,I2,1H/,I2,1H/,I2,4X,F10.2,4X,F10.2,6X,F6.2,2X,F10.2)       YES
   40 CONTINUE                                                             YES
   50 CONTINUE                                                             YES
C                                                                          YES
C     WRITE RECORD TO OUTPUT FILE                                          YES
      WRITE(YESOUT)IID,IREGID,IZONID,ISTRID,YSTR,                          YES
     1  (IZPRDD(J),YSCI(J),VSYCI(J),J=1,6)                                 YES
      NYESR=NYESR+1                                                        YES
C                                                                          YES
C     READ NEXT RECORD                                                     YES
      IF(IREG.EQ.ENDR.AND.IZONE.EQ.ENDZ)IEND=1                             YES
      READ(YESERR)COUN,IREG,IZONE,ISTRAT,YTRUE,                            YES
     1  (IZULU(I),BIAS(I),SD(I),I=1,6)                                     YES
      NREC(5)=NREC(5)+1                                                    YES
      IF(ENDZ.EQ.0.AND.COUN .NE. ZZZZ) GO TO 22                            YES
      IF(ENDZ.EQ.0.AND.COUN .EQ.ZZZZ) GO TO 60                             YES
      IF(IEND.EQ.0.AND.COUN.EQ.4HZZZZ) GO TO 55                            YES
      IF((IEND.EQ.1).AND.(IREG.NE.ENDR.OR.IZONE.NE.ENDZ)                   YES
     1  .AND.COUN.NE.4HZZZZ) GO TO 55                                      YES
      IF(COUN.EQ.4HZZZZ) GO TO 60                                          YES
      GO TO 22                                                             YES
C                                                                          YES
C      CANNOT FIND ENDING ZONE                                             YES
   55 CONTINUE                                                             YES
C      REPORT ERROR                                                        YES
       CALL ERRMES(3HYES,3HYES,2,0)                                        YES
   60 CONTINUE                                                             YES
C                                                                          YES
C      WRITE TRAILER RECORD TO OUTPUT FILE AND CLEAN UP                    YES
       NREC(5)= NREC(5)-1                                                  YES
       ITEMP=LYESO-1                                                       YES
       WRITE(YESOUT)ZZZZ,(IFILL,I=1,ITEMP)                                 YES
```

```
      REWIND YESOUT                                                     YFS
      REWIND YESERR                                                     YFS
      RFTURN                                                            YFS
      END                                                               YFS
```

```
@ FLT YSUB,1,760427, 39100     , 1

      FUNCTION YSUB (AA,B)                                              YSUB
C        COMPUTES THE QUANTITY Y USED IN THE CONFIDENCE LEVEL           YSUB
C        CALCULATIONS.                                                  YSUB
C                                                                       YSUB
C     CALLING SEQUENCE PARAMETERS ...                                   YSUB
C        AA = VARIANCE                                                  YSUB
C        B  = REFERENCE VALUE                                           YSUB
C                                                                       YSUB
C        ARGUMENT LIST FOR ERROR PROCESSING                             ARGLST
      COMMON  /ARGLST/                                                  ARGLST
     1   NERRS ,NFATAL,NPERRS,NARG  ,ARG(10)                            ARGLST
      DIMENSION  IARG(10)                                               ARGLST
      EQUIVALENCE  ( IARG,ARG )                                         ARGLST
C                                                                       ARGLST
C                                                                       YSUB
      DATA  ICTR  / 0 /                                                 YSUB
C                                                                       YSUB
C                                                                       YSUB
      A= AA                                                             YSUB
      IF ( A .GT. 0.0 )  GO TO 120                                      YSUB
C                                                                       YSUB
C        A IS NEGATIVE OR ZERO.                                         YSUB
      IF ( -A .LT. 1.E-7*B )  GO TO 110                                 YSUB
      ARG(3)= A                                                         YSUB
      ARG(4)= B                                                         YSUB
      ICTR= ICTR + 1                                                    YSUB
      IF ( ICTR .LT. 6 )  CALL ERRMES (3HCAS,4HYSUB,18,0)               YSUB
  110 A= 0.0                                                            YSUB
C                                                                       YSUB
  120 YSUB= AMAX1 ( SQRT(A),1.E-30 )                                    YSUB
C                                                                       YSUB
  900 RETURN                                                            YSUB
      END                                                               YSUB

 11. TOC
```